Hadoop HDFS / HDFS-6581 Write to single replica in memory / HDFS-6934

Move checksum computation off the hot path when writing to RAM disk

Details

Type:	Sub-task	Status:	CLOSED
Priority:	Major	Resolution:	Fixed
Affects Version/s:	None	Fix Version/s:	2.6.0
Component/s:	datanode, hdfs-client		
Labels:	None		
Target Version/s:	2.6.0		
Hadoop Flags:	Reviewed		

Description

Since local RAM is considered reliable we can avoid writing checksums on the hot path when replicas are being written to a local RAM disk.

The checksum can be computed by the lazy writer when moving replicas to disk.

Attachments

h6934_20141003b.patch	29 kB	04/Oct/14 02:49
h6934_20141005.patch	50 kB	05/Oct/14 23:20
HDFS-6934.3.patch	85 kB	24/Oct/14 19:29
HDFS-6934.4.patch	118 kB	25/Oct/14 06:55
HDFS-6934.5.patch	119 kB	26/Oct/14 05:09
HDFS-6934-branch-2.6.5.patch	119 kB	27/Oct/14 19:22

Activity

Tsz-wo Sze added a comment - 03/Oct/14 20:24

Currently, client computes checksum and sends it to the datanodes. Datanode verifies the checksum and then persists it with the block data.

In this JIRA, we propose that client does not compute checksum anymore for in-memory replica in order to give a better performance. The checksum will be computed when the datanode lazy persists the replica to disk.

Tsz-wo Sze added a comment - 04/Oct/14 01:09

We have the following cases in datanode side when receiving a packet:

No Checksum Received	Writing to Transient Storage	Action
T	T	No checksum is stored
T	F	Compute checksum and write to disk
F	T	Write checksum to memory
F	F	Write checksum to disk

Arpit Agarwal added a comment - 04/Oct/14 01:55

Hi Nicholas, the proposed approach of computing checksum in DN when write to memory is requested sounds fine to me. The truth table also looks good.

Perhaps the only concern could be that the CPU usage will show up in the DN process and not the client process, but the overall CPU usage of the node will not be affected as long as we ensure memory writes are always local. I filed ~~HDFS-7192~~ for that, will fix it later.

Tsz-wo Sze added a comment - 04/Oct/14 02:31

Thanks Arpit for the comment. The CPU usage in DN won't be increased:

- For single replica case, if client sends checksum, then DN will verify it (i.e. DN needs to compute checksum). If client does not send checksum, DN will compute it when persisting the replica.
- For multiple replica case, client must send checksum since only the first DN could write to memory. The other DNs still need checksum.

Tsz-wo Sze added a comment - 04/Oct/14 02:49

h6934_20141003b.patch: changed DN. Still needs to change DFSClient and adds tests.

Hadoop QA added a comment - 04/Oct/14 03:23

-1 overall. Here are the results of testing the latest attachment
http://issues.apache.org/jira/secure/attachment/12672901/h6934_20141003b.patch
against trunk revision c8212ba.

+1 @author. The patch does not contain any @author tags.

+1 tests included. The patch appears to include 1 new or modified test files.

+1 javac. The applied patch does not increase the total number of javac compiler warnings.

+1 javadoc. There were no new javadoc warning messages.

+1 eclipse:eclipse. The patch built with eclipse:eclipse.

-1 findbugs. The patch appears to cause Findbugs (version 2.0.3) to fail.

+1 release audit. The applied patch does not increase the total number of release audit warnings.

+1 core tests. The patch passed unit tests in .

+1 contrib tests. The patch passed contrib unit tests.

Test results: https://builds.apache.org/job/PreCommit-HDFS-Build/8322//testReport/
Console output: https://builds.apache.org/job/PreCommit-HDFS-Build/8322//console

This message is automatically generated.

Tsz-wo Sze added a comment - 05/Oct/14 23:20

h6934_20141005.patch: changes client side to not compute checksum for the single replica writing to memory case.

Hadoop QA added a comment - 06/Oct/14 02:15

-1 overall. Here are the results of testing the latest attachment
http://issues.apache.org/jira/secure/attachment/12673037/h6934_20141005.patch
against trunk revision 16333b4.

+1 @author. The patch does not contain any @author tags.

-1 tests included. The patch doesn't appear to include any new or modified tests.
Please justify why no new tests are needed for this patch.
Also please list what manual steps were performed to verify this patch.

-1 javac. The applied patch generated 1318 javac compiler warnings (more than the trunk's current 1268 warnings).

+1 javadoc. There were no new javadoc warning messages.

+1 eclipse:eclipse. The patch built with eclipse:eclipse.

-1 findbugs. The patch appears to introduce 1 new Findbugs (version 2.0.3) warnings.

+1 release audit. The applied patch does not increase the total number of release audit warnings.

-1 core tests. The patch failed these unit tests in hadoop-hdfs-project/hadoop-hdfs:

org.apache.hadoop.hdfs.server.namenode.ha.TestPipelinesFailover
org.apache.hadoop.hdfs.server.namenode.ha.TestFailureToReadEdits
org.apache.hadoop.hdfs.TestEncryptionZonesWithKMS

The following test timeouts occurred in hadoop-hdfs-project/hadoop-hdfs:

org.apache.hadoop.fs.TestUrlStreamHandler

+1 contrib tests. The patch passed contrib unit tests.

Test results: https://builds.apache.org/job/PreCommit-HDFS-Build/8325//testReport/
Findbugs warnings: https://builds.apache.org/job/PreCommit-HDFS-Build/8325//artifact/patchprocess/newPatchFindbugsWarningshadoop-hdfs.html
Javac warnings: https://builds.apache.org/job/PreCommit-HDFS-Build/8325//artifact/patchprocess/diffJavacWarnings.txt
Console output: https://builds.apache.org/job/PreCommit-HDFS-Build/8325//console

This message is automatically generated.

Jing Zhao added a comment - 07/Oct/14 22:47

The patch looks pretty good to me. Some comments and questions:

1. Maybe we should list the changes that have been made in the patch in the jira. Considering the patch changes both the client and DN's code, that will make it easier for us to revisit the jira in the future.
2. Do we also need to handle the checksum for short circuit read?
3. It will be better to have a javadoc in `BlockSender#BlockSender` to explain idea behind the length check.

```
-          checksumIn = new DataInputStream(
-              new BufferedInputStream(metaIn, HdfsConstants.IO_FILE_BUFFER_SIZE));
+          if (metaIn.getLength() > BlockMetadataHeader.getHeaderSize()) {
+            checksumIn = new DataInputStream(new BufferedInputStream(
+                metaIn, HdfsConstants.IO_FILE_BUFFER_SIZE));
```

4. It will be great to have more unit tests to cover both read and write scenarios.

Chris Nauroth added a comment - 24/Oct/14 19:29

I'm resuming the work started on this by Nicholas. I'm uploading patch v3. I believe this addresses the prior feedback from Jing. Here is a summary of the changes in the current patch.

`FSOutputSummer`, `Options`, `DataChecksum`: Refactoring to help simplify usage of the checksum classes in HDFS.
`BlockReaderFactory`: Pass along storage type for use later by individual block reader classes.
`BlockReaderLocal`: Allow skipping checksums if the replica is on transient storage. However, do not anchor, because the intent is not that clients should prevent the DataNode from evicting a replica from RAM.
`BlockReaderLocalLegacy`: Skip checksums when reading a replica on transient storage. Do not cache path information for a replica on transient storage, because eviction at the DataNode may later invalidate that cached path. Unlike the newer short-circuit read implementation, we don't have a communication channel (the shared memory segment) for the DataNode to communicate path invalidation back to the client, so we don't have a more graceful way to handle this.
`DFSClient`: Minor changes associated with refactoring of the checksum classes in Common.
`DFSInputStream`: Get storage type from the `LocatedBlock`. Pass it along to `BlockReaderFactory`.
`DFSOutputStream`: Do not write checksums if using transient storage. Minor changes associated with refactoring of the checksum classes in Common.
`LocatedBlock`: Change `toString` to aid debugging.
`BlockMetadataHeader`: Similar to the changes in Common, this is refactoring to make code related to checksum handling simpler.
`BlockReceiver`: Do not require writing checksums on transient storage.
`BlockSender`: Do not verify checksum when reading a replica from transient storage.
`ReplicaInPipeline`: Minor cleanups.
`ReplicaOutputStreams`: Expose whether or not the replica is on transient storage.
`BlockPoolSlice`: Minor change to go along with the refactoring of checksum classes.
`FsDatasetImpl`: During lazy persist, instead of copying a meta file from transient storage, we now calculate the checksum to create a new meta file. Also, when a replica gets evicted from transient storage, make a call to the `ShortCircuitRegistry` to inform clients that they must evict from their caches.
`RamDiskAsyncLazyPersistService`: Log stack trace for lazy persist failures to aid debugging.
`RamDiskReplicaLruTracker`: Minor clean-up of raw generic type.
`TestLazyPersistFiles`: New tests added covering short-circuit read (new and legacy) in combination with eviction, including corruption detection. Refactored to use a global timeout setting.

Xiaoyu Yao added a comment - 24/Oct/14 22:40

Thanks cnauroth for working on this. The fix looks good to me. Below are some of my comments:

1. FSOutputSummer.java
If FSOutputSummer#getChecksumSize() is added to reduce individual call sum.getChecksumSize() in FSOutputSummer, there are still three Sum.getChecksumSize() calls that can be replaced with getChecksumSize().

2. DataChecksum.java
NIT: Can you add spaces between operators of * and + in DataChecksum#getChecksumSize().

3. BlockReceiver.java
If you put Line 582~583 before Line 574, you can use

```
If (shouldNotWriteChecksum)
```

instead of duplicating the same logic.

4. DFSInputStream.java
Line 954: This can be moved out of the loop
StorageType[] storageTypes = block.getStorageTypes();

Line 955: Can you add comment on the relationship between index of nodes ⓘ to the storage type selection
if (storageTypes != null && i < storageTypes.length)
{ storageType = storageTypes[i]; }

5. TestLazyPersistFiles.java
We have a separate test class for SCR related tests in TestScrLazyPersistFiles.java so that the temporary domain socket does not need to be created for non-SCR cases. Can put the new SCR cases there?

I notice the test timeout attribute is removed. Is there a particular reason for that?

DFS_DATANODE_RAM_DISK_LOW_WATERMARK_REPLICAS has been changed to DFS_DATANODE_RAM_DISK_LOW_WATERMARK_BYTES by ~~HDFS-6988~~, can you sync with the trunk and update the patch?

Hadoop QA added a comment - 24/Oct/14 22:48

-1 overall. Here are the results of testing the latest attachment
http://issues.apache.org/jira/secure/attachment/12676980/HDFS-6934.3.patch
against trunk revision 86ac0d4.

+1 @author. The patch does not contain any @author tags.

+1 tests included. The patch appears to include 2 new or modified test files.

+1 javac. The applied patch does not increase the total number of javac compiler warnings.

+1 javadoc. There were no new javadoc warning messages.

+1 eclipse:eclipse. The patch built with eclipse:eclipse.

-1 findbugs. The patch appears to introduce 1 new Findbugs (version 2.0.3) warnings.

+1 release audit. The applied patch does not increase the total number of release audit warnings.

-1 core tests. The patch failed these unit tests in hadoop-common-project/hadoop-common hadoop-hdfs-project/hadoop-hdfs:

org.apache.hadoop.hdfs.protocolPB.TestPBHelper
org.apache.hadoop.hdfs.server.balancer.TestBalancer
org.apache.hadoop.hdfs.TestDFSClientRetries

+1 contrib tests. The patch passed contrib unit tests.

Test results: https://builds.apache.org/job/PreCommit-HDFS-Build/8526//testReport/
Findbugs warnings: https://builds.apache.org/job/PreCommit-HDFS-Build/8526//artifact/patchprocess/newPatchFindbugsWarningshadoop-hdfs.html
Console output: https://builds.apache.org/job/PreCommit-HDFS-Build/8526//console

This message is automatically generated.

Chris Nauroth added a comment - 25/Oct/14 06:55

Thank you for reviewing, Xiaoyu. Lots of good suggestions. Here is patch v4, with the following changes.

FSOutputSummer: Used the common getChecksumSize method in more places like you suggested.
DataChecksum: Added whitespace between operators.
DFSClient: Fixed minor Findbugs warning identified in last Jenkins run.
DFSInputStream: In getBestNodeDNAddrPair, moved storage types outside the loop and added comment explaining indexed lookup of storage type.
LazyWriterTestCase: There was a lot of code duplication for helper methods in TestLazyPersistFiles and TestScrLazyPersistFiles. I took the opportunity to refactor into a new abstract base class that they can share.
TestScrLazyPersistFiles: The new tests are now in here.

> If you put Line 582~583 before Line 574, you can use...

I don't think we can make this change. The two conditions are logically slightly different. On line 574:

```
      if (checksumReceivedLen == 0 && !streams.isTransientStorage()) {
```

On line 582:

```
      final boolean shouldNotWriteChecksum = checksumReceivedLen == 0
          && streams.isTransientStorage();
```

Note the negation of isTransientStorage() in the first one. The first case is for detecting if a write to non-transient storage did not send a checksum, and therefore it needs to be calculated here. The second case is for detecting if this is a write to transient storage, and therefore the checksum needs to be skipped entirely.

> I notice the test timeout attribute is removed. Is there a particular reason for that?

The patch is using the Timeout JUnit rule to specify a timeout applied across every test case in one place instead of specifying the timeout in each Test annotation. We've been using this approach where possible in new tests, because it avoids duplication of the timeout values and helps prevent introducing new tests that forgot to specify a timeout. In the current version of the patch, the Timeout rule is in the abstract base class, which covers it for all tests in both subclasses.

Hadoop QA added a comment - 25/Oct/14 10:18

-1 overall. Here are the results of testing the latest attachment
http://issues.apache.org/jira/secure/attachment/12677086/HDFS-6934.4.patch
against trunk revision f44cf99.

+1 @author. The patch does not contain any @author tags.

+1 tests included. The patch appears to include 4 new or modified test files.

+1 javac. The applied patch does not increase the total number of javac compiler warnings.

+1 javadoc. There were no new javadoc warning messages.

+1 eclipse:eclipse. The patch built with eclipse:eclipse.

+1 findbugs. The patch does not introduce any new Findbugs (version 2.0.3) warnings.

+1 release audit. The applied patch does not increase the total number of release audit warnings.

-1 core tests. The patch failed these unit tests in hadoop-common-project/hadoop-common hadoop-hdfs-project/hadoop-hdfs:

org.apache.hadoop.hdfs.TestLeaseRecovery2
org.apache.hadoop.hdfs.protocolPB.TestPBHelper
org.apache.hadoop.hdfs.TestDFSClientRetries

+1 contrib tests. The patch passed contrib unit tests.

Test results: https://builds.apache.org/job/PreCommit-HDFS-Build/8540//testReport/
Console output: https://builds.apache.org/job/PreCommit-HDFS-Build/8540//console

This message is automatically generated.

Tsz-wo Sze added a comment - 26/Oct/14 04:21

cnauroth, thanks for the big help on continuing this work. The new patch looks great. Some minor comments:

- Actually, storage types are always present. For a new client talking to an old clusters, the storage types will be filled with default type, i.e. DISK; see PBHelper.convertStorageTypes(List<StorageTypeProto> storageTypesList, int expectedSize).

```
+          // Storage types are ordered to correspond with nodes, so use the same
+          // index to get storage type.  Storage types might not be present.  For
+          // example, the client might be communicating with a cluster running a
+          // prior version before implementation of storage types.
+          if (storageTypes != null && i < storageTypes.length) {
+            storageType = storageTypes[i];
+          }
```

- I suggest to add the information of block, nodes and ignoredNodes to the exception.

```
+    if (chosenNode == null) {
+      throw new IOException("No live nodes contain current block");
+    }
```

- DFSOutputStream.initLazyPersist(..) is not removed in the new patch. Also, the hard coded string "LAZY_PERSIST" in isLazyPersist(..) should be replaced by HdfsConstants.MEMORY_STORAGE_POLICY_NAME, which was introduced recently.

Chris Nauroth added a comment - 26/Oct/14 05:09

Thank you for the review, Nicholas. Here is patch v5 with your suggestions. I also added null checks in `LocatedBlock#toString` to address the test failures reported by Jenkins.

Hadoop QA added a comment - 26/Oct/14 08:29

+1 overall. Here are the results of testing the latest attachment
http://issues.apache.org/jira/secure/attachment/12677153/HDFS-6934.5.patch
against trunk revision 65d95b1.

+1 @author. The patch does not contain any @author tags.

+1 tests included. The patch appears to include 4 new or modified test files.

+1 javac. The applied patch does not increase the total number of javac compiler warnings.

+1 javadoc. There were no new javadoc warning messages.

+1 eclipse:eclipse. The patch built with eclipse:eclipse.

+1 findbugs. The patch does not introduce any new Findbugs (version 2.0.3) warnings.

+1 release audit. The applied patch does not increase the total number of release audit warnings.

+1 core tests. The patch passed unit tests in hadoop-common-project/hadoop-common hadoop-hdfs-project/hadoop-hdfs.

+1 contrib tests. The patch passed contrib unit tests.

Test results: https://builds.apache.org/job/PreCommit-HDFS-Build/8543//testReport/
Console output: https://builds.apache.org/job/PreCommit-HDFS-Build/8543//console

This message is automatically generated.

Xiaoyu Yao added a comment - 26/Oct/14 23:11

The 5th patch looks pretty good to me. Thanks cnauroth for addressing all the questions.
+1 (non-binding).

Tsz-wo Sze added a comment - 27/Oct/14 05:15

+1 patch looks good. Thanks Chris!

Hudson added a comment - 27/Oct/14 16:51

FAILURE: Integrated in Hadoop-trunk-Commit #6352 (See https://builds.apache.org/job/Hadoop-trunk-Commit/6352/)
~~HDFS-6934~~. Move checksum computation off the hot path when writing to RAM disk. Contributed by Chris Nauroth. (cnauroth: rev 463aec11718e47d4aabb86a7a539cb973460aae6)

- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BlockSender.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/RamDiskAsyncLazyPersistService.java
- hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt
- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/util/Shell.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSClient.java

- hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/SimulatedFSDataset.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BlockMetadataHeader.java
- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/Options.java
- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FSOutputSummer.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/RamDiskReplicaLruTracker.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSOutputStream.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/BlockPoolSlice.java
- hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/util/TestShell.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/BlockReaderFactory.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/BlockReaderLocal.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/BlockReaderLocalLegacy.java
- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/io/nativeio/NativeIO.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/ReplicaOutputStreams.java
- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/util/DataChecksum.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/ReplicaInPipeline.java
- hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/TestScrLazyPersistFiles.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/LocatedBlock.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BlockReceiver.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsDatasetImpl.java
- hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/LazyPersistTestCase.java
- hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/TestLazyPersistFiles.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSInputStream.java

---

Chris Nauroth added a comment - 27/Oct/14 19:06

There was a very slight merge conflict on merging to branch-2.6. It was just an indentation difference caused by the fact that the tracing work (~~HDFS-7055~~) is in branch-2, but not branch-2.6.

I have attached a branch-2.6 patch. szetszwo, are you +1 for this version too? The only difference compared to branch-2 is some indentation in `BlockReaderLocal`.

Hudson added a comment - 27/Oct/14 19:22

FAILURE: Integrated in Hadoop-trunk-Commit #6356 (See https://builds.apache.org/job/Hadoop-trunk-Commit/6356/)
~~HDFS-6934~~. Revert files accidentally committed. (cnauroth: rev 5b1dfe78b8b06335bed0bcb83f12bb936d4c021b)

- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/util/Shell.java
- hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/util/TestShell.java
- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/io/nativeio/NativeIO.java

---

Chris Nauroth added a comment - 27/Oct/14 19:22

Reattaching the correct patch file for branch-2.6.

Tsz-wo Sze added a comment - 27/Oct/14 19:27

+1 the 2.6 patch looks good.

> There was a very slight merge conflict on merging to branch-2.6. ...

In such case, we should cherry-pick the trunk commit and resolve the merge manually.

Chris Nauroth added a comment - 27/Oct/14 20:01

I have committed this to trunk, branch-2 and branch-2.6. Xiaoyu and Nicholas, thank you for the code reviews.

> In such case, we should cherry-pick the trunk commit and resolve the merge manually.

Yes, this is how I handled it, so you'll see it show up in the branch-2.6 commit log as a cherry-pick. The merge conflict was just enough that I thought it would be best to seek another +1 before pushing it up. Thanks again!

Hudson added a comment - 28/Oct/14 11:50

SUCCESS: Integrated in Hadoop-Yarn-trunk #726 (See https://builds.apache.org/job/Hadoop-Yarn-trunk/726/)
~~HDFS-6934~~. Move checksum computation off the hot path when writing to RAM disk. Contributed by Chris Nauroth. (cnauroth: rev 463aec11718e47d4aabb86a7a539cb973460aae6)

- hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSInputStream.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/BlockReaderLocal.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSOutputStream.java
- hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/util/TestShell.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BlockReceiver.java
- hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/TestLazyPersistFiles.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/BlockPoolSlice.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSClient.java
- hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/TestScrLazyPersistFiles.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BlockSender.java
- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/Options.java
- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/util/Shell.java
- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/io/nativeio/NativeIO.java
- hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/LazyPersistTestCase.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/ReplicaOutputStreams.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/BlockReaderFactory.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BlockMetadataHeader.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/RamDiskReplicaLruTracker.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/RamDiskAsyncLazyPersistService.java
- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FSOutputSummer.java
- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/util/DataChecksum.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/LocatedBlock.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/ReplicaInPipeline.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/BlockReaderLocalLegacy.java
- hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/SimulatedFSDataset.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsDatasetImpl.java
~~HDFS-6934~~. Revert files accidentally committed. (cnauroth: rev 5b1dfe78b8b06335bed0bcb83f12bb936d4c021b)
- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/io/nativeio/NativeIO.java
- hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/util/TestShell.java
- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/util/Shell.java

Hudson added a comment - 28/Oct/14 13:35

FAILURE: Integrated in Hadoop-Mapreduce-trunk #1940 (See https://builds.apache.org/job/Hadoop-Mapreduce-trunk/1940/)
~~HDFS-6934~~. Move checksum computation off the hot path when writing to RAM disk. Contributed by Chris Nauroth. (cnauroth: rev 463aec11718e47d4aabb86a7a539cb973460aae6)

- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/util/Shell.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSClient.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/BlockReaderFactory.java
- hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/TestScrLazyPersistFiles.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/RamDiskAsyncLazyPersistService.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSOutputStream.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/LocatedBlock.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/BlockReaderLocal.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/ReplicaOutputStreams.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsDatasetImpl.java
- hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/util/TestShell.java

- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/io/nativeio/NativeIO.java
- hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/TestLazyPersistFiles.java
- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/util/DataChecksum.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BlockMetadataHeader.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/ReplicaInPipeline.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSInputStream.java
- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FSOutputSummer.java
- hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/LazyPersistTestCase.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/RamDiskReplicaLruTracker.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BlockReceiver.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/BlockReaderLocalLegacy.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BlockSender.java
- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/Options.java
- hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/BlockPoolSlice.java
- hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/SimulatedFSDataset.java

~~HDFS-6934~~. Revert files accidentally committed. (cnauroth: rev 5b1dfe78b8b06335bed0bcb83f12bb936d4c021b)

- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/util/Shell.java
- hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/util/TestShell.java
- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/io/nativeio/NativeIO.java

Hudson added a comment - 28/Oct/14 14:21

SUCCESS: Integrated in Hadoop-Hdfs-trunk #1915 (See https://builds.apache.org/job/Hadoop-Hdfs-trunk/1915/)
~~HDFS-6934~~. Move checksum computation off the hot path when writing to RAM disk. Contributed by Chris Nauroth. (cnauroth: rev 463aec11718e47d4aabb86a7a539cb973460aae6)

- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/RamDiskAsyncLazyPersistService.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/ReplicaOutputStreams.java
- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/Options.java
- hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/TestLazyPersistFiles.java
- hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/LazyPersistTestCase.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BlockMetadataHeader.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/BlockReaderFactory.java
- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/io/nativeio/NativeIO.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BlockSender.java
- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/util/DataChecksum.java
- hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/SimulatedFSDataset.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSClient.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/ReplicaInPipeline.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/BlockReaderLocalLegacy.java
- hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt
- hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/util/TestShell.java
- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/util/Shell.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/BlockPoolSlice.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSOutputStream.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/LocatedBlock.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSInputStream.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsDatasetImpl.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/BlockReaderLocal.java
- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FSOutputSummer.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BlockReceiver.java
- hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/TestScrLazyPersistFiles.java
- hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/RamDiskReplicaLruTracker.java

~~HDFS-6934~~. Revert files accidentally committed. (cnauroth: rev 5b1dfe78b8b06335bed0bcb83f12bb936d4c021b)

- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/io/nativeio/NativeIO.java
- hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/util/TestShell.java
- hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/util/Shell.java

People

Assignee:

Chris Nauroth

Reporter:

Arpit Agarwal

Votes:

0 Vote for this issue

Watchers:

7 Start watching this issue

Dates

Created:

23/Aug/14 21:35

Updated:

17/Jun/15 03:33

Resolved:

27/Oct/14 20:01