

**git\_comments:**

1. \* \* Licensed to the Apache Software Foundation (ASF) under one \* or more contributor license agreements. See the NOTICE file \* distributed with this work for additional information \* regarding copyright ownership. The ASF licenses this file \* to you under the Apache License, Version 2.0 (the \* "License"); you may not use this file except in compliance \* with the License. You may obtain a copy of the License at \* \* <http://www.apache.org/licenses/LICENSE-2.0> \* \* Unless required by applicable law or agreed to in writing, software \* distributed under the License is distributed on an "AS IS" BASIS, \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. \* See the License for the specific language governing permissions and \* limitations under the License.
2. \* \* RequestLog object for use with Http
3. \* \* Log4j Appender adapter for HttpRequestLog
4. \* \* Licensed to the Apache Software Foundation (ASF) under one \* or more contributor license agreements. See the NOTICE file \* distributed with this work for additional information \* regarding copyright ownership. The ASF licenses this file \* to you under the Apache License, Version 2.0 (the \* "License"); you may not use this file except in compliance \* with the License. You may obtain a copy of the License at \* \* <http://www.apache.org/licenses/LICENSE-2.0> \* \* Unless required by applicable law or agreed to in writing, software \* distributed under the License is distributed on an "AS IS" BASIS, \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. \* See the License for the specific language governing permissions and \* limitations under the License.
5. \* \* A dummy invocation handler extending RetryInvocationHandler. It drops the \* first N number of responses. This invocation handler is only used for testing.
6. \* \* Licensed to the Apache Software Foundation (ASF) under one \* or more contributor license agreements. See the NOTICE file \* distributed with this work for additional information \* regarding copyright ownership. The ASF licenses this file \* to you under the Apache License, Version 2.0 (the \* "License"); you may not use this file except in compliance \* with the License. You may obtain a copy of the License at \* \* <http://www.apache.org/licenses/LICENSE-2.0> \* \* Unless required by applicable law or agreed to in writing, software \* distributed under the License is distributed on an "AS IS" BASIS, \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. \* See the License for the specific language governing permissions and \* limitations under the License.
7. \* \* Licensed to the Apache Software Foundation (ASF) under one \* or more contributor license agreements. See the NOTICE file \* distributed with this work for additional information \* regarding copyright ownership. The ASF licenses this file \* to you under the Apache License, Version 2.0 (the \* "License"); you may not use this file except in compliance \* with the License. You may obtain a copy of the License at \* \* <http://www.apache.org/licenses/LICENSE-2.0> \* \* Unless required by applicable law or agreed to in writing, software \* distributed under the License is distributed on an "AS IS" BASIS, \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. \* See the License for the specific language governing permissions and \* limitations under the License.
8. \* \* Licensed to the Apache Software Foundation (ASF) under one \* or more contributor license agreements. See the NOTICE file \* distributed with this work for additional information \* regarding copyright ownership. The ASF licenses this file \* to you under the Apache License, Version 2.0 (the \* "License"); you may not use this file except in compliance \* with the License. You may obtain a copy of the License at \* \* <http://www.apache.org/licenses/LICENSE-2.0> \* \* Unless required by applicable law or agreed to in writing, software \* distributed under the License is distributed on an "AS IS" BASIS, \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. \* See the License for the specific language governing permissions and \* limitations under the License.
9. \* \* Total number of elements in the list.
10. \* \* Default initial size is 6 elements, since typical minimum object \* size is 64 bytes, and this leaves enough space for the object \* header.
11. \* \* @param initialChunkCapacity the capacity of the first chunk to be \* allocated \* @param maxChunkSize the maximum size of any chunk allocated
12. \* \* Cache of the last element in the 'chunks' array above. \* This speeds up the add operation measurably.
13. **comment:** \* \* Simplified List implementation which stores elements as a list \* of chunks, each chunk having a maximum size. This improves over \* using an ArrayList in that creating a large list will never require \* a large amount of contiguous heap space -- thus reducing the likelihood \* of triggering a CMS compaction pause due to heap fragmentation. \* \* The first chunks allocated are small, but each additional chunk is \* 50% larger than the previous, ramping up to a configurable maximum \* chunk size. Reasonable defaults are provided which should be a good \* balance between not making any large allocations while still retaining \* decent performance. \* \* This currently only supports a small subset of List operations -- \* namely addition and iteration.  
**label:** code-design
14. \* \* The maximum number of elements for any chunk.

15. \* Licensed to the Apache Software Foundation (ASF) under one \* or more contributor license agreements. See the NOTICE file \* distributed with this work for additional information \* regarding copyright ownership. The ASF licenses this file \* to you under the Apache License, Version 2.0 (the \* "License"); you may not use this file except in compliance \* with the License. You may obtain a copy of the License at \* \* <http://www.apache.org/licenses/LICENSE-2.0> \* \* Unless required by applicable law or agreed to in writing, software \* distributed under the License is distributed on an "AS IS" BASIS, \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. \* See the License for the specific language governing permissions and \* limitations under the License.
16. \* \* The capacity with which the last chunk was allocated.
17. \* \* Default max size is 8K elements - which, at 8 bytes per element \* should be about 64KB -- small enough to easily fit in contiguous \* free heap space even with a fair amount of fragmentation.
18. \* \* The chunks which make up the full list.
19. \* \* The capacity of the first chunk to allocate in a cleared list.
20. Insert a bunch of elements.
21. \* Licensed to the Apache Software Foundation (ASF) under one \* or more contributor license agreements. See the NOTICE file \* distributed with this work for additional information \* regarding copyright ownership. The ASF licenses this file \* to you under the Apache License, Version 2.0 (the \* "License"); you may not use this file except in compliance \* with the License. You may obtain a copy of the License at \* \* <http://www.apache.org/licenses/LICENSE-2.0> \* \* Unless required by applicable law or agreed to in writing, software \* distributed under the License is distributed on an "AS IS" BASIS, \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. \* See the License for the specific language governing permissions and \* limitations under the License.
22. test ChunkedArrayList
23. Check that it got chunked.
24. \* \* Licensed to the Apache Software Foundation (ASF) under one \* or more contributor license agreements. See the NOTICE file \* distributed with this work for additional information \* regarding copyright ownership. The ASF licenses this file \* to you under the Apache License, Version 2.0 (the \* "License"); you may not use this file except in compliance \* with the License. You may obtain a copy of the License at \* \* <http://www.apache.org/licenses/LICENSE-2.0> \* \* Unless required by applicable law or agreed to in writing, software \* distributed under the License is distributed on an "AS IS" BASIS, \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. \* See the License for the specific language governing permissions and \* limitations under the License.
25. \* \* Get {@link ContainerId} of the container being initialized or stopped. \* \* @return the container ID
26. \* \* Get {@link Resource} the resource capability allocated to the container \* being initialized or stopped. \* \* @return the resource capability.
27. \* \* Base context class for {@link AuxiliaryService} initializing and stopping a \* container.
28. \* \* Licensed to the Apache Software Foundation (ASF) under one \* or more contributor license agreements. See the NOTICE file \* distributed with this work for additional information \* regarding copyright ownership. The ASF licenses this file \* to you under the Apache License, Version 2.0 (the \* "License"); you may not use this file except in compliance \* with the License. You may obtain a copy of the License at \* \* <http://www.apache.org/licenses/LICENSE-2.0> \* \* Unless required by applicable law or agreed to in writing, software \* distributed under the License is distributed on an "AS IS" BASIS, \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. \* See the License for the specific language governing permissions and \* limitations under the License.
29. \* \* Get user of the container being initialized or stopped. \* \* @return the user
30. \* \* Licensed to the Apache Software Foundation (ASF) under one \* or more contributor license agreements. See the NOTICE file \* distributed with this work for additional information \* regarding copyright ownership. The ASF licenses this file \* to you under the Apache License, Version 2.0 (the \* "License"); you may not use this file except in compliance \* with the License. You may obtain a copy of the License at \* \* <http://www.apache.org/licenses/LICENSE-2.0> \* \* Unless required by applicable law or agreed to in writing, software \* distributed under the License is distributed on an "AS IS" BASIS, \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. \* See the License for the specific language governing permissions and \* limitations under the License.
31. \* \* Initialization context for {@link AuxiliaryService} when starting a \* container. \*
32. \* \* Licensed to the Apache Software Foundation (ASF) under one \* or more contributor license agreements. See the NOTICE file \* distributed with this work for additional information \* regarding copyright ownership. The ASF licenses this file \* to you under the Apache License, Version 2.0 (the \* "License"); you may not use this file except in compliance \* with the License. You may obtain a copy of the License at \* \* <http://www.apache.org/licenses/LICENSE-2.0> \* \* Unless required by applicable law or agreed to in writing, software \* distributed under the License is distributed on an "AS IS" BASIS, \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. \* See the License for the specific language governing permissions and \* limitations under the License.

33. \* Termination context for {@link AuxiliaryService} when stopping a \* container. \*
34. \* Licensed to the Apache Software Foundation (ASF) under one \* or more contributor license agreements. See the NOTICE file \* distributed with this work for additional information \* regarding copyright ownership. The ASF licenses this file \* to you under the Apache License, Version 2.0 (the \* "License"); you may not use this file except in compliance \* with the License. You may obtain a copy of the License at \* \* <http://www.apache.org/licenses/LICENSE-2.0> \* Unless required by applicable law or agreed to in writing, software \* distributed under the License is distributed on an "AS IS" BASIS, \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. \* See the License for the specific language governing permissions and \* limitations under the License.
35. \* Test that the server with request logging enabled \* \* @throws Throwable on failure
36. This case is used for testing.
37. The number of NN response dropped by client proactively in each RPC call. For testing NN retry cache, we can set this property with positive value.
38. HA case
39. \* Generate a dummy namenode proxy instance that utilizes our hacked \* {@link LossyRetryInvocationHandler}. Proxy instance generated using this \* method will proactively drop RPC responses. Currently this method only \* support HA setup. IllegalStateException will be thrown if the given \* configuration is not for HA. \* \* @param config the configuration containing the required IPC \* properties, client failover configurations, etc. \* @param nameNodeUri the URI pointing either to a specific NameNode \* or to a logical nameservice. \* @param xface the IPC interface which should be created \* @param numResponseToDrop The number of responses to drop for each RPC call \* @return an object containing both the proxy and the associated \* delegation token service it corresponds to \* @throws IOException if there is an error creating the proxy
40. \* Default connection factory may be overridden in tests to use smaller timeout values
41. \* Shutdown all the nodes in the cluster.
42. \* Containers preempted by the framework.
43. \* A container is finishing on this NodeManager. This is a signal to this \* {@link AuxiliaryService} about the same. \* \* @param stopContainerContext context for the container termination
44. \* A new container is started on this NodeManager. This is a signal to \* this {@link AuxiliaryService} about the container initialization. \* This method is called when the NodeManager receives the container launch \* command from the ApplicationMaster and before the container process is \* launched. \* \* @param initContainerContext context for the container's initialization
45. if the current state is NEW it means the CONTAINER\_INIT was never sent for the event, thus no need to send the CONTAINER\_STOP
46. \* End of Active services
47. \* "Active" services. Services that need to run only on the Active RM. \* These services are managed (initialized, started, stopped) by the \* {@link CompositeService} RMActiveServices. \* \* RM is active when (1) HA is disabled, or (2) HA is enabled and the RM is \* in Active state.
48. the Exception from stateStore.init() needs to be handled for HA and we need to give up master status if we got fenced
49. \* RMActiveServices handles all the Active services in the RM.
50. the Exception from loadState() needs to be handled for HA and we need to give up master status if we got fenced
51. get a token so the client can communicate with the app attempt NOTE: token may be unavailable if the attempt is not running
52. \* Create a token for authenticating a client connection to the app attempt \* @param clientName the name of the client requesting the token \* @return the token or null if the attempt is not running
53. There must be at least one container allocated, because a CONTAINER\_ALLOCATED is emitted after an RMContainer is constructed, and is put in SchedulerApplication#newlyAllocatedContainers. Then, YarnScheduler#allocate will fetch it.
54. \* Utility to create a {@link ContainerStatus} during exceptional \* circumstances. \* \* @param containerId {@link ContainerId} of returned/released/lost container. \* @param diagnostics diagnostic message \* @return `ContainerStatus` for an returned/released/lost \* container
55. \* Utility to create a {@link ContainerStatus} during exceptional \* circumstances. \* \* @param containerId {@link ContainerId} of returned/released/lost container. \* @param diagnostics diagnostic message \* @return `ContainerStatus` for an returned/released/lost \* container
56. Use ConcurrentSkipListMap because applications need to be ordered
57. At RM restart it is safe to assume that all the previously added tokens are valid
58. Delegation token renewal is delayed until ClientRMService starts. As it is required to short circuit the token renewal calls.
59. Renewing token and adding it to timer calls are separated purposefully If user provides incorrect token then it should not be added for renewal.

60. stateQuery is deprecated.
61. do nothing
62. do nothing
63. kill the app attempt and verify client token is unavailable
64. Imitating the thread of rmappattempt that will get the app
65. Imitating the thread of scheduler that will add and remove apps

#### git\_commits:

1. **summary:** Merging r1520450 through r1521565 from trunk to branch HDFS-2832  
**message:** Merging r1520450 through r1521565 from trunk to branch HDFS-2832 git-svn-id: <https://svn.apache.org/repos/asf/hadoop/common/branches/HDFS-2832@1521566> 13f79535-47bb-0310-9956-ffa450edef68

#### github\_issues:

#### github\_issues\_comments:

#### github\_pulls:

#### github\_pulls\_comments:

#### github\_pulls\_reviews:

#### jira\_issues:

1. **summary:** Enable support for heterogeneous storages in HDFS - DN as a collection of storages  
**description:** HDFS currently supports configuration where storages are a list of directories. Typically each of these directories correspond to a volume with its own file system. All these directories are homogeneous and therefore identified as a single storage at the namenode. I propose, change to the current model where Datanode \* is a \* storage, to Datanode \* is a collection \* of storages.  
**label:** code-design
2. **summary:** Enable support for heterogeneous storages in HDFS - DN as a collection of storages  
**description:** HDFS currently supports configuration where storages are a list of directories. Typically each of these directories correspond to a volume with its own file system. All these directories are homogeneous and therefore identified as a single storage at the namenode. I propose, change to the current model where Datanode \* is a \* storage, to Datanode \* is a collection \* of storages.  
**label:** code-design
3. **summary:** Enable support for heterogeneous storages in HDFS - DN as a collection of storages  
**description:** HDFS currently supports configuration where storages are a list of directories. Typically each of these directories correspond to a volume with its own file system. All these directories are homogeneous and therefore identified as a single storage at the namenode. I propose, change to the current model where Datanode \* is a \* storage, to Datanode \* is a collection \* of storages.
4. **summary:** Enable support for heterogeneous storages in HDFS - DN as a collection of storages  
**description:** HDFS currently supports configuration where storages are a list of directories. Typically each of these directories correspond to a volume with its own file system. All these directories are homogeneous and therefore identified as a single storage at the namenode. I propose, change to the current model where Datanode \* is a \* storage, to Datanode \* is a collection \* of storages.
5. **summary:** Enable support for heterogeneous storages in HDFS - DN as a collection of storages  
**description:** HDFS currently supports configuration where storages are a list of directories. Typically each of these directories correspond to a volume with its own file system. All these directories are homogeneous and therefore identified as a single storage at the namenode. I propose, change to the current model where Datanode \* is a \* storage, to Datanode \* is a collection \* of storages.
6. **summary:** Enable support for heterogeneous storages in HDFS - DN as a collection of storages  
**description:** HDFS currently supports configuration where storages are a list of directories. Typically each of these directories correspond to a volume with its own file system. All these directories are homogeneous and therefore identified as a single storage at the namenode. I propose, change to the current model where Datanode \* is a \* storage, to Datanode \* is a collection \* of storages.
7. **summary:** Enable support for heterogeneous storages in HDFS - DN as a collection of storages  
**description:** HDFS currently supports configuration where storages are a list of directories. Typically each of these directories correspond to a volume with its own file system. All these directories are homogeneous and therefore identified as a single storage at the namenode. I propose, change to the current model where Datanode \* is a \* storage, to Datanode \* is a collection \* of storages.

- [illegible]

- [illegible]











therefore identified as a single storage at the namenode. I propose, change to the current model where Datanode \* is a \* storage, to Datanode \* is a collection \* of storages.

- [illegible]

therefore identified as a single storage at the namenode. I propose, change to the current model where Datanode \* is a \* storage, to Datanode \* is a collection \* of storages.

- [illegible]

- therefore identified as a single storage at the namenode. I propose, change to the current model where Datanode \* is a \* storage, to Datanode \* is a collection \* of storages.
104. **summary:** Enable support for heterogeneous storages in HDFS - DN as a collection of storages  
**description:** HDFS currently supports configuration where storages are a list of directories. Typically each of these directories correspond to a volume with its own file system. All these directories are homogeneous and therefore identified as a single storage at the namenode. I propose, change to the current model where Datanode \* is a \* storage, to Datanode \* is a collection \* of storages.
  105. **summary:** Enable support for heterogeneous storages in HDFS - DN as a collection of storages  
**description:** HDFS currently supports configuration where storages are a list of directories. Typically each of these directories correspond to a volume with its own file system. All these directories are homogeneous and therefore identified as a single storage at the namenode. I propose, change to the current model where Datanode \* is a \* storage, to Datanode \* is a collection \* of storages.
  106. **summary:** Enable support for heterogeneous storages in HDFS - DN as a collection of storages  
**description:** HDFS currently supports configuration where storages are a list of directories. Typically each of these directories correspond to a volume with its own file system. All these directories are homogeneous and therefore identified as a single storage at the namenode. I propose, change to the current model where Datanode \* is a \* storage, to Datanode \* is a collection \* of storages.
  107. **summary:** Enable support for heterogeneous storages in HDFS - DN as a collection of storages  
**description:** HDFS currently supports configuration where storages are a list of directories. Typically each of these directories correspond to a volume with its own file system. All these directories are homogeneous and therefore identified as a single storage at the namenode. I propose, change to the current model where Datanode \* is a \* storage, to Datanode \* is a collection \* of storages.
  108. **summary:** Enable support for heterogeneous storages in HDFS - DN as a collection of storages  
**description:** HDFS currently supports configuration where storages are a list of directories. Typically each of these directories correspond to a volume with its own file system. All these directories are homogeneous and therefore identified as a single storage at the namenode. I propose, change to the current model where Datanode \* is a \* storage, to Datanode \* is a collection \* of storages.
  109. **summary:** Enable support for heterogeneous storages in HDFS - DN as a collection of storages  
**description:** HDFS currently supports configuration where storages are a list of directories. Typically each of these directories correspond to a volume with its own file system. All these directories are homogeneous and therefore identified as a single storage at the namenode. I propose, change to the current model where Datanode \* is a \* storage, to Datanode \* is a collection \* of storages.
  110. **summary:** Enable support for heterogeneous storages in HDFS - DN as a collection of storages  
**description:** HDFS currently supports configuration where storages are a list of directories. Typically each of these directories correspond to a volume with its own file system. All these directories are homogeneous and therefore identified as a single storage at the namenode. I propose, change to the current model where Datanode \* is a \* storage, to Datanode \* is a collection \* of storages.
  111. **summary:** Enable support for heterogeneous storages in HDFS - DN as a collection of storages  
**description:** HDFS currently supports configuration where storages are a list of directories. Typically each of these directories correspond to a volume with its own file system. All these directories are homogeneous and therefore identified as a single storage at the namenode. I propose, change to the current model where Datanode \* is a \* storage, to Datanode \* is a collection \* of storages.

#### jira\_issues\_comments:

1. **body:** Advantages: # Support for heterogeneous storages: ## DN could support along with disks, other types of storage such as flash etc. ## Suitable storage can be chosen based on client preference such as need for random reads etc. # Block report scaling: instead of a single monolithic block report, a smaller block report per storage becomes possible. This is important with the growth in disk capacity and number of disks per datanode. # Better granularity of storage failure handling: ## DN could just indicate loss of storage and namenode can handle it better since it knows the list of blocks belonging to a storage. ## DN could locally handle storage failures or provide decommissioning of a storage by marking a storage as ReadOnly. # Hot pluggability of disks/storages: adding and deleting a storage to a node is simplified. # Other flexibility: includes future enhancements to balance storages within a datanode, balancing the load (number of transceivers) per storage etc and better block placement strategies. Backward compatibility: The existing grouping of all storages under a single storage ID is a specific case of the generalized model proposed above. This change will be backward compatible with the existing deployments.  
**label:** code-design
2. **body:** HDFS-2880 as part of HDFS-2832 changed the abstraction from DataNode as single storage to DataNode as a collection of storages. The proposal is to use this to build support for heterogeneous, tiered storage support in HDFS. Currently we are considering three tiers of storage: # Disk # SSD # Memory In order to achieve this, we will start with the following changes: # Storages in DataNodes will have storage type. #

DataNode in block report sends the list of storages along with the type and list of blocks corresponding to each storage. # NameNode will track storages and storage types reported by the DataNode. Appropriate changes will be made to DataNode maps and block maps to do this in the NameNode. # Clients can provide preference for storage type to write to. # Block locations given to the client will indicate in addition to DataNode location, a storage type. With storage type corresponding to memory, we could explore writing data to memory storage tier. Also for faster reads data extra copies of data could be done in memory storage tier thus facilitating caching of the data. This proposal is very high level. We will post a design document soon with more details. This work is significant and we intend do this in multiple phases. We will open Jiras to track these issues and folks interested in contributing can collaborate on this work. Please reach out to me. -- Suresh Srinivas, Sanjay Radia and Arpit Agarwal

**label:** code-design

3. Doesn't the above repurposing of this JIRA overlap substantially with HDFS-4672? We already opened that JIRA to see if folks may be interested in collaborating on tiered storage support. I'm curious why that hasn't happened substantially there but there is this post on this issue today. Many of the items on the above list could have been taken from HDFS-4672.
4. {quote} We will open Jiras to track these issues and folks interested in contributing can collaborate on this work. Please reach out to me. - Suresh Srinivas, Sanjay Radia and Arpit Agarwal {quote} As you know there is an email from me reaching out to you on exactly the topic of tiered storage in your inbox still waiting for a response. I know conclusively at least Suresh and Sanjay received a copy. I sent that at the invitation of one of your colleagues last month. Perhaps I can conclude from the silence that email is an incorrect way to "reach out", in which case I would welcome your participation on HDFS-4762 in a truly inclusive community process of collaboration, rather than dubious invitations, based on my personal experience, here.
5. I wonder how these tiers interact with alternative storage services? For instance, I run my cluster in EC2 and want a replica stored on S3. Where does this use case fit in?
6. bq. Many of the items on the above list could have been taken from HDFS-4672. Wow. Please go read the description. I will make it easy by adding a link here - <https://issues.apache.org/jira/browse/HDFS-2832?focusedCommentId=13192326&page=com.atlassian.jira.plugin.system.issuetabpanels:comment-tabpanel#comment-13192326>. The description while does not spell details, gives an idea of how far sweeping this proposal is. Some of the tasks were prioritized such as making Datanode as a collection storages because they would require significant protocol changes. After that we have slowed down. However this jira can be worked on in multiple independent phases. bq. ... I would welcome your participation on HDFS-4762 in a truly inclusive community process of collaboration, rather than dubious invitations, based on my personal experience, here. So you sending an email on a public mailing list is true inclusive community process. However, a similar comment on this jira is not!!!! I have hard time understanding this. In fact many of the features done in HDFS have had meetups with not only HDFS folks, but also higher stack components such as HBase and experienced Hadoop Ops. So please stop these kinds of comments that you have no way participating in this. Just because you create a jira with additional information, does not make it original jira. It is still a duplicate. Spirit of collaboration that you so strongly talk about could have been demonstrated by just posting a comment on this jira instead of opening a new one. In fact I had told you in person in Hadoop Summit Europe that we will start working on this in April (which we could not, given snapshot feature). I am going to create some of the sub-tasks that I have always been thinking about. I will also leave it open, for people to comment on, do code reviews, and contribute patches to. Hopefully that should help you to participate in this work, if for some reason you did not feel involved already. Again if your frustration is stemming from the fact that you already have code for some of this, we could consider them adding as part of subtasks that is being created.
7. Andrew, I was always confused about the creation of HDFS-4672 given that HDFS-2832 existed for quite a while and with a bunch of subtasks complete. note: Konstantine asked you about the duplication in April "What is the difference between this issue and HDFS-2832?" in [comment | <https://issues.apache.org/jira/browse/HDFS-4672?focusedCommentId=13629403&page=com.atlassian.jira.plugin.system.issuetabpanels:comment-tabpanel#comment-13629403>] to which you have not responded to date. However noting the description and comments on HDFS-4672 it is clear that HDFS-4672 seems to be focusing on policies for placement. I resigned to letting this Jira (HDFS-2832) focus on the raw mechanisms and HDFS-4672 focus on the policies; as a result I did not raise the issue about you duplicating this jira when you created HDFS-4672. I did that out of politeness and moving forward but perhaps that was a mistake on my part. Also note that I have commented on the other jira and have not ignored it. Your recent tweet about this before giving me a chance to respond is unfortunate, especially given that HDFS-4672 is overlapping with HDFS-2832. - isn't this the pot calling the kettle black? Getting back to the business building Hadoop: We need to move quickly forward by adding mechanism in HDFS to allow different kinds of storage while the longer discussions on policies continue. Given that a recent Jira on ram caching (HDFS-4949) is planning to move quickly, it is important that we start by providing the the needed mechanisms. We plan to add additional mechanisms via sub-jiras which can support HDFS-4949. You are welcome to contribute with patches and discussions. Similarly we plan to continue to participate in the longer ranged discussions on policies and also contribute with patches.

8. bq. I wonder how these tiers interact with alternative storage services? For instance, I run my cluster in EC2 and want a replica stored on S3. Where does this use case fit in? Currently we are considering the following: local Disk, remote {local rack|remote rack} disk, local SSD, remote {local rack|remote rack} SSD, local RAM, remote {local rack|remote rack} RAM, RAID, tape, remote storage (such as S3, SAN) etc. We are also considering temporary/permanent replicas etc. Will post some details next week. We can discuss if it includes all the storage types we should be considering.
9. I have created couple of jiras, thus dumping out stuff I have been carrying in my head (and other that I have discussed this with) for a long time. These set of jiras enable support for heterogeneous storages in different tiers. I would like to start working on HDFS-4985 immediately as this might also be useful for expressing datanode cache in HDFS-4949 in a way that is compatible with the proposal here.
10. **body:** Is there an overall design document that one can follow? HDFS-2802 was great in this regard with initial docs followed by revisions.  
**label:** documentation
11. [~sanjay.radia] bq. Your recent tweet about this before giving me a chance to respond is unfortunate, especially given that HDFS-4672 is overlapping with HDFS-2832. - isn't this the pot calling the kettle black? This argumentation rather than hearing those concerns, the scope increase here in the first place that ignored HDFS-4672 - and now in addition it is a "duplicate", and the unanswered private communication all indicate that there is some kind of competitive dynamic here that is not helpful. I prefer to defuse the situation and move on rather than engage in some kind of debate club atmosphere. Therefore today I resolved HDFS-4672 as Later. Please see my comment at the tail of it.
12. [~bikassaha] We'll post a design doc by next week.
13. Hey Arpit, like Bikas I'd also love to see a design doc. I'm also potentially interested in picking up some of the outstanding subtasks, since they seem very related to HDFS-4949.
14. There may not be lot of overlap between this and HDFS-4949, other than indicating storage type as memory in block report. That said you are welcome to pickup the tasks.
15. **body:** Hi [~andrew.wang], that's great to hear. I got sidetracked into some retry cache related changes for 2.1.0-beta. I will post the doc next week.  
**label:** documentation
16. Here is a design doc for this feature. We will be creating a new HDFS-2832 branch for this work. We plan to split the feature into more sub-tasks and work on it in two phases: # Heterogeneous storage support in HDFS # APIs to expose support to applications Any feedback on the document is welcome. We can schedule a meeting to have a discussion on the feature some time in late August or early September depending on the interest.
17. Just have created a new branch: <http://svn.apache.org/viewvc/hadoop/common/branches/HDFS-2832/>
18. < To support Storage Types, each DataNode must be treated as a collection of storages. (excerpt from pdf) Consider a cluster with a set of DataNodes with high end hardware (eg: SSD), and another set of DataNodes with low end hardware (eg: HDD). Each datanode is homogenous by itself, but the cluster itself is heterogeneous. Can the user still specify storage preference using StorageType and get expected results?
19. John, {quote}Can the user still specify storage preference using StorageType and get expected results? {quote} We don't make any assumptions about the cluster layout. The storages attached to a DataNode may be of the same of different types.
20. Nice doc! If there are multiple hard drives on a data node will they be represented as a single storage volume of type HDD. Will it be legal to have multiple storage volumes of the same type on the same data node? Say 1 volume per HDD.
21. Hi Bikas, each storage directory/volume (we used the terms interchangeably) will be represented as a distinct DataNodeStorage. They will not be grouped by types.
22. Sorry I am still not clear on this. Let me re-phrase. If there are 10 disks on a data node, will it be legal to create 2 volumes of type HDD on that Datanode with 5 disks each? I am guessing that the volume+type list for a datanode will come from config. i.e. the datanode will not be figuring this out automatically by inspecting the hardware on the machine.
23. Yes, directory+type will be read from the configuration (sec 5.2.1). A volume corresponds to a single storage directory, so if you have 10 unstriped disks there will be 10 at least storage volumes.
24. **body:** I finally got a chance to read this doc, nice work. A few questions: - For quota management, have you considered the YARN-like abstraction of users and pools? We're moving down that avenue in HDFS-4949, and it'd be nice to eventually have a single abstraction if we can. I get that for a first cut, it's easier to stick with the existing disk quota system. - How do you expect applications to handle runtime failures? If I have a stream open and my write fails due to lack of SSD quota, can I change it to retry the write to HDD? Do I get metrics so I can alert somewhere? - How do you handle block migration of files opened by long-lived applications like HBase that also use short-circuit local reads? Let's say HBase initially writes all its files to SSD, then we want to periodically migrate them to HDD. HBase holds onto the SSD file descriptors indefinitely, preventing reclamation of SSD capacity. - If "File Storage Preferences" are part of file metadata, what happens when the files are copied, or distcp'd to another cluster? - Why do we want a default "Storage Preferences" specified on a directory? I'd actually prefer if we make applications explicitly request special treatment when they open a

stream. - Let's say I'm a cluster operator, and have nodes with both PCI-e and SATA SSDs. Can I differentiate between them? How about if I add nodes with an unknown StorageType like NVRAM? Basically: what's required to add a new StorageType? - Also related, when I bring up a new StorageType in my cluster, how do I make my applications start using it? Do I need to submit patches to HBase to now know how to use NVRAM properly? This seems like one of the downsides of physical storage types, logical means apps can do this more automatically.

**label:** code-design

25. Nice doc. My additional comments is: we should consider the case that remote storage that fall into the same failure group. i.e. Storage Volumes of Node A, Node B is on NAS-a, so only 1 replica should be placed on these two nodes although they may on different racks.
26. **body:** Hi Andrew, Thanks for looking at doc, great questions. {quote} - For quota management, have you considered the YARN-like abstraction of users and pools? We're moving down that avenue in HDFS-4949, and it'd be nice to eventually have a single abstraction if we can. I get that for a first cut, it's easier to stick with the existing disk quota system. {quote} I am interested in seeing how the pools abstraction is defined and whether it can cover all our use cases. Do you have a design doc? We chose this approach because it extends the existing quota system and APIs and more importantly covers our use cases. {quote} - How do you expect applications to handle runtime failures? If I have a stream open and my write fails due to lack of SSD quota, can I change it to retry the write to HDD? Do I get metrics so I can alert somewhere? {quote} "Out of quota" is a hard failure just like hitting the disk space quota limit. The application must change the Storage Preference on the file to continue. We have not discussed metrics yet. {quote} - How do you handle block migration of files opened by long-lived applications like HBase that also use short-circuit local reads? Let's say HBase initially writes all its files to SSD, then we want to periodically migrate them to HDD. HBase holds onto the SSD file descriptors indefinitely, preventing reclamation of SSD capacity. {quote} Quota should be blocked indefinitely until the files can be moved off their current Storage Type. We did not cover this use case, so thanks for calling it out! I will make the update. {quote} - If "File Storage Preferences" are part of file metadata, what happens when the files are copied, or distcp'd to another cluster? {quote} I think this is a tools decision. We probably want to lose the File Attributes, with an option to preserve them. We will document this in more detail when we get to updating the tools. {quote} - Why do we want a default "Storage Preferences" specified on a directory? I'd actually prefer if we make applications explicitly request special treatment when they open a stream. {quote} Storage Preferences are not supported on directories. Please let me know if you see anything in the doc implying otherwise and I will fix it. {quote} - Let's say I'm a cluster operator, and have nodes with both PCI-e and SATA SSDs. Can I differentiate between them? How about if I add nodes with an unknown StorageType like NVRAM? Basically: what's required to add a new StorageType? - Also related, when I bring up a new StorageType in my cluster, how do I make my applications start using it? Do I need to submit patches to HBase to now know how to use NVRAM properly? This seems like one of the downsides of physical storage types, logical means apps can do this more automatically. {quote} Adding a new StorageType needs code and update to the StorageType enum. We made the trade-off for API and implementation simplicity for v1 but we are not ruling out adding support for logical classification in the future.

**label:** code-design

27. bq. I am interested in seeing how the pools abstraction is defined... I've got a little writeup in the HDFS-4949 design doc, but we're still figuring it out. Arun expressed interest in reviewing our RM plan, so you could ask him about pools/etc too. bq. The application must change the Storage Preference on the file to continue. Is this a stream API? The doc only mentions specifying storage preferences at create time via {{DFSCClient#create}}. I'm wondering how this works for the case where I write to SSD, run out of capacity, then want write the rest of my file to HDD. Do I need to close the file, modify the Storage Preference, then reopen for append? This also potentially requires migrating the last block to HDD, since storage types are tracked per-block, and then you might hit the "HBase keeps fds open forever" issue. It'd be nice to see something in the doc discussing the above, under construction replicas are hard :) bq. Storage Preferences are not supported on directories. Please let me know if you see anything in the doc implying otherwise and I will fix it. Grep for this line: "Storage preferences could be specified per-file or per-directory."
28. Andrew, {quote} Grep for this line: "Storage preferences could be specified per-file or per-directory." {quote} I meant to say we considered both options but chose to go with per-file preferences. I should have worded it better. :) {quote} Is this a stream API? The doc only mentions specifying storage preferences at create time via DFSCClient#create. {quote} This is briefly mentioned in section 6.4 - \_File Attribute APIs to query, set and clear file attributes which will be used to modify Storage Preferences\_. We have describe File Attributes in detail before we start working on the feature API. {quote} I'm wondering how this works for the case where I write to SSD, run out of capacity, then want write the rest of my file to HDD. Do I need to close the file, modify the Storage Preference, then reopen for append? This also potentially requires migrating the last block to HDD, since storage types are tracked per-block, and then you might hit the "HBase keeps fds open forever" issue. {quote} We differentiate between running out of capacity and running out of quota (7.1, 1b and 1c). HDFS



- handles `_Out of Capacity_` transparently by allocating subsequent blocks on HDD as fallback and does not require any migration to make forward progress.
29. Hi [~junping\_du], thanks for looking at the doc. {quote} My additional comments is: we should consider the case that remote storage that fall into the same failure group. i.e. Storage Volumes of Node A, Node B is on NAS-a, so only 1 replica should be placed on these two nodes although they may on different racks. {quote} We have not considered this use case. Are you running multiple DataNodes over the same NAS for redundancy? Please feel free to file a Jira with your feature idea and motivation and we can discuss how to proceed after we have initial level of support for Heterogeneous Storages. Sound good?
30. **body:** Thanks for quickly reply. [~arpitagarwal]. bq. We have not considered this use case. Are you running multiple DataNodes over the same NAS for redundancy? I have two use cases in my mind: 1. As one kind of DR solution, user can choose to put 1 or 2 replica on a remote reliable storage (i.e. NAS backed with SAN). Multiple DNs connect to NAS will have more bandwidth. 2. In virtualization case, some virtual machines are backed with cluster FS (i.e. VMFS) on shared storage rather than local disks (it may not be the most cost-effective way but not corner case in enterprise environment). Some DNs on VMs could be backed with same shared storage. bq. Please feel free to file a Jira with your feature idea and motivation and we can discuss how to proceed after we have initial level of support for Heterogeneous Storages. Sound good? Yes. That make sense. Will file JIRA later.  
**label:** code-design
31. **body:** Just read through the design doc. Thanks for writing it up. One thing I noticed under tooling that is not covered is extensions to the "fs -du" and/or "fs -count" command. Given that SSD is typically a very constrained resource, I imagine that administrators will want to be able to see where the space usage is going on a directory hierarchy level, in order to chase down users who are using too much (or applications which accidentally are specifying all their files should be on SSD). Similarly, what's the planned way in which existing FsShell commands that write data will be extended to specify these policies? For example, if I want to upload a file with "hadoop fs -put", how can I specify that it should go on SSD? Maybe I missed it, but are there other CLI extensions planned in order to retrieve and set the file attributes necessary to migrate data?  
**label:** code-design
32. Todd, thanks for reviewing and the feedback. FsShell extensions to query and set storage policies are mentioned in sec 8,2 and agreed about the remaining two also. I'll mention them when I rev the design doc.
33. Arpit and team - nice document. A few questions and comments: # Does the design allow a `_single_ datanode` to have multiple replicas of a block (presumably each on different storage types - e.g. SSD and HDD)? If so `_(and I think it should)_`, this would seem to require some refactoring of the `{{FSDatasetInterface}}` which is oriented around the fact that a block maps to a single volume (e.g. `{{FSVolume getVolume(Block b)}}`). # Does the design intend to allow for Storage types (e.g. RAM) to be backed by non-file-addressable stores? If so `_(and I think it should)_`, this would also require some redesign of some areas: #\* The `{{FSDatasetInterface}}` abstraction which allows for pluggable (i.e. non-file-addressable) block storage mechanisms is global to the DataNode. Perhaps it should be pluggable on a `_per-storage_` basis - e.g. having a `{{MemoryFSDataset}}` and a `{{FileFSDataset}}` implementation co-existing within a single DataNode instance. Thinking about this some more, this might also help address my point above re: `{{FSDatasetInterface}}` being oriented around a single `{{FSVolume}}` per block. #\* Areas where File-addressable block access is assumed outside the `{{FSDatasetInterface}}` abstraction: #\*\* `{{BlockSender}}` / `{{BlockReceiver}}` which downcast to `FileInputStreams` to obtain the underlying FD #\*\* `{{DataStorage.linkBlocks}}` which uses hardlinks for upgrade/revert scenarios
34. **body:** Thanks for the feedback Eric. Both of these would be good to have and came up during design discussions but we have not addressed either. For #2, in addition to your points there are other locations where storage directories are assumed to be File-addressable. I am not sure of the amount of work involved here. Multiple replicas per Datanode looks easier and can be done on top of the Heterogeneous Storage work. We would need phase 1 of the feature to support multiple storages.  
**label:** code-design
35. h2832\_20131023.patch: for Jenkins testing.
36. {color:red}-1 overall{color}. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12609899/h2832\\_20131023.patch](http://issues.apache.org/jira/secure/attachment/12609899/h2832_20131023.patch) against trunk revision . {color:red}-1 patch{color}. The patch command could not apply the patch. Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5267/console> This message is automatically generated.
37. h2832\_20131023b.patch: generated from the project root.
38. h5417\_20131025.patch: updated with HDFS-5417.
39. {color:red}-1 overall{color}. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12610345/h5417\\_20131025.patch](http://issues.apache.org/jira/secure/attachment/12610345/h5417_20131025.patch) against trunk revision . {color:red}-1 patch{color}. The patch command could not apply the patch. Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5278/console> This message is automatically generated.
40. {color:red}-1 overall{color}. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12610345/h5417\\_20131025.patch](http://issues.apache.org/jira/secure/attachment/12610345/h5417_20131025.patch) against trunk revision .

- {color:red}-1 patch{color}. The patch command could not apply the patch. Console output:  
<https://builds.apache.org/job/PreCommit-HDFS-Build/5279//console> This message is automatically generated.
41. > h5417\_20131025.patch: updated with HDFS-5417. Oops, the file should be h2832\_20131025.patch.
42. The Jenkins failed to run through result, rename the patch and submit it again.
43. {color:red}-1 overall{color}. Here are the results of testing the latest attachment  
[http://issues.apache.org/jira/secure/attachment/12610761/h2832\\_20131028b.patch](http://issues.apache.org/jira/secure/attachment/12610761/h2832_20131028b.patch) against trunk revision .  
 {color:red}-1 patch{color}. The patch command could not apply the patch. Console output:  
<https://builds.apache.org/job/PreCommit-HDFS-Build/5302//console> This message is automatically generated.
44. We need to merge with HDFS-4949 changes that got integrated into trunk last night. Looking at it.
45. h5452\_20131103.patch
46. should be h2832\_20131103.patch.
47. **body:** Hey guys, I was wondering if we really need to change storageID to UUID. I thought that the storageID approach that `_each DN is able to generate a unique id independently of the others_` is a good feature to retain. UUID as you noted is not unique and needs to be coordinated through NameNode. I understand you have multiple storages on the same DN, and you need unique ids independently of the ip, and port. # They should be unique with existing implementation of `{{createNewStorageId()}}`. `{code}storageid = <random, ip, port, currentTime>{code}` If you generate ids sequentially one after another, currentTime should be different. It can be replaced by nano-time if id generation is done in different threads. # You can also add to storageID an attribute that characterizes the disk volume or the directory as a new component. Examples of the new attribute could be disk serial number, or the storage directory inode number. It seems that introduction of UUIDs was unnecessary, unless of course I missed some context.
- label:** code-design
48. h5457\_20131104.patch
49. Oops, wrong file again. Sorry for the noise. Should be h2832\_20131104.patch
50. Konstantin, {quote} I thought that the storageID approach that each DN is able to generate a unique id independently of the others is a good feature to retain. {quote} Storage (UU)IDs are independently generated on the Datanode in `{{DataStorage#format}}`. {quote} UUID as you noted is not unique and needs to be coordinated through NameNode. {quote} Not true. `{{UUID#randomUUID}}` generates RFC-4122 compliant UUIDs which are unique for all practical purposes without NameNode coordination. {quote} You can also add to storageID an attribute that characterizes the disk volume or the directory as a new component. Examples of the new attribute could be disk serial number, or the storage directory inode number. It seems that introduction of UUIDs was unnecessary, unless of course I missed some context. {quote} Part of the rationale is in HDFS-5115. Making them UUIDs simplifies the generation logic. Decoupling them from volume/directory characteristics allows future storage media that do not have a disk serial number or inode number.
51. **body:** > `UUID#randomUUID` generates RFC-4122 compliant UUIDs which are unique \*for all practical purposes\* RFC-4122 has a special note about "distributed applications". But let's just think about it in general. `randomUUID` is based on pseudo random sequence of numbers, which is like a Mobius Strip or just a loop. It actually works well if you generate IDs on a single node, because the sequence lasts long without repetitions. In our case we initiate thousands of pseudo random sequences (one per node), each starting from a random number. Let's mark those starting numbers on the Mobius Strip or the loop. Then we actually decreased the probability of uniqueness because now in order to get a collision one of the nodes need to reach the starting point of another node, rather than going all around the loop. So in distributed environment we increase the probability of collision with each new node added. And when you add more storage types per node you further increase the collision probability. "for all practical purposes" as I understand it in the case means that probability of non-unique IDs is low. But it does not mean impossible. The consequences of a storageID collision are pretty bad, hard to detect and recover. At the same time `{{DataNode.createNewStorageId()}}` generates unique IDs as of today. Why changing it to a problematic approach? > Part of the rationale is in HDFS-5115. Making them UUIDs simplifies the generation logic. Looks like HDFS-5115 was based on an incomplete assumption: bq. The Storage ID is currently generated from the DataNode's IP+Port+Random components while in fact it also includes currentTime, which guarantees the uniqueness of ids generated on the same node, unless somebody resets the machine clock to the past.
- label:** code-design
52. Konstantin, UUID generation uses a cryptographically secure PRNG. On Linux this is `/dev/random`, the fallback is SHA1PRNG with a period of  $2^{160}$ . With a billion nodes the probability of a collision in a 128-bit space is less than 1 in  $10^{20}$ . Note that what was previously the storageID is now the datanode UUID and it is generated once for the lifetime of a datanode.
53. Arpit, I think we just agreed that collisions among UUIDs are possible but have low probability. This is a concern for me. Even though unlikely, a collision if it happens creates a serious problem for the system integrity. Does it concern you? In my previous comment I tried to explain that in distributed case the randomness of it is the main problem. Forget for a moment about PRNGs. Assume that UUID is an incremental counter (such as generation stamp (and now block id)), which is incremented by each node independently but at start up each chooses a randomly number to start from. On a single node ++ can go on without collisions for a long enough

time to guarantee I will never see it. Y4K bug is fine with me. But if you take the second node and randomly choose a starting number it could be close to (1000 apart) the starting point of the first node. Then the second node can only generate 1000 storageIDs before colliding with those generated by the other node. The same is with PRNG you just replace ++ with next(). Long period doesn't matter if you choose your starting points randomly.

- [illegible]

- patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HDFS-Build/5384/testReport/> Release audit warnings: <https://builds.apache.org/job/PreCommit-HDFS-Build/5384/artifact/trunk/patchprocess/patchReleaseAuditProblems.txt> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HDFS-Build/5384/artifact/trunk/patchprocess/newPatchFindbugsWarningshadoop-hdfs.html> Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5384/console> This message is automatically generated.
59. Attaching editsStored for {{TestOfflineEditsViewer}}.
60. {color:red}-1 overall{color}. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12613434/h2832\\_20131112.patch](http://issues.apache.org/jira/secure/attachment/12613434/h2832_20131112.patch) against trunk revision . {color:red}-1 patch{color}. The patch command could not apply the patch. Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5408/console> This message is automatically generated.
61. {color:red}-1 overall{color}. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12613511/h2832\\_20131112b.patch](http://issues.apache.org/jira/secure/attachment/12613511/h2832_20131112b.patch) against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 43 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:red}-1 findbugs{color}. The patch appears to introduce 1 new Findbugs (version 1.3.9) warnings. {color:red}-1 release audit{color}. The applied patch generated 1 release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-hdfs-project/hadoop-hdfs: org.apache.hadoop.hdfs.tools.offlineEditsViewer.TestOfflineEditsViewer The following test timeouts occurred in hadoop-hdfs-project/hadoop-hdfs: org.apache.hadoop.hdfs.TestDFSStartupVersions org.apache.hadoop.hdfs.server.namenode.TestListCorruptFileBlocks {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HDFS-Build/5414/testReport/> Release audit warnings: <https://builds.apache.org/job/PreCommit-HDFS-Build/5414/artifact/trunk/patchprocess/patchReleaseAuditProblems.txt> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HDFS-Build/5414/artifact/trunk/patchprocess/newPatchFindbugsWarningshadoop-hdfs.html> Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5414/console> This message is automatically generated.
62. The unit test failure may due to no cleanup as I still see old "48" version in edit xml in Jenkins test log (my local env is OK) Filed HDFS-5510 to fix a findbug warning. The audit warning can be ignored.
63. **body:** TestOfflineEditsViewer is due to missing binary file editsStored. TestListCorruptFileBlocks looks like spurious timeout, I cannot duplicate the failure. Looking at TestDFSStartupVersions.  
**label:** code-design
64. > The probability of no collision is  $P = n! / ((n-m)! n^m)$ . Nicholas, your math is good, as usually. It works well for random events. My point though is about the deterministic nature of pseudo random sequences. In the sense that if you initiate two independent PRNGs with the same seed, they will generate exactly the same sequences. Sorry for late reply, travelling.
65. {color:red}-1 overall{color}. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12613979/h2832\\_20131114.patch](http://issues.apache.org/jira/secure/attachment/12613979/h2832_20131114.patch) against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 44 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:red}-1 release audit{color}. The applied patch generated 1 release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-hdfs-project/hadoop-hdfs: org.apache.hadoop.hdfs.server.blockmanagement.TestUnderReplicatedBlocks org.apache.hadoop.hdfs.server.namenode.TestListCorruptFileBlocks org.apache.hadoop.hdfs.tools.offlineEditsViewer.TestOfflineEditsViewer {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HDFS-Build/5440/testReport/> Release audit warnings: <https://builds.apache.org/job/PreCommit-HDFS-Build/5440/artifact/trunk/patchprocess/patchReleaseAuditProblems.txt> Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5440/console> This message is automatically generated.
66. {quote} My point though is about the deterministic nature of pseudo random sequences. In the sense that if you initiate two independent PRNGs with the same seed, they will generate exactly the same sequences. {quote} Seed collision is even less likely than outright UUID collision. The seed space for SHA1PRNG is  $2^{160}$ .
67. {color:red}-1 overall{color}. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12614486/h2832\\_20131118.patch](http://issues.apache.org/jira/secure/attachment/12614486/h2832_20131118.patch) against trunk revision . {color:red}-1 patch{color}. Trunk compilation may be broken. Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5464/console> This message is automatically generated.

68. {color:red}-1 overall{color}. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12614486/h2832\\_20131118.patch](http://issues.apache.org/jira/secure/attachment/12614486/h2832_20131118.patch) against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 45 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:red}-1 release audit{color}. The applied patch generated 1 release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-hdfs-project/hadoop-hdfs: org.apache.hadoop.hdfs.server.blockmanagement.TestUnderReplicatedBlocks org.apache.hadoop.hdfs.tools.offlineEditsViewer.TestOfflineEditsViewer {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HDFS-Build/5473/testReport/> Release audit warnings: <https://builds.apache.org/job/PreCommit-HDFS-Build/5473/artifact/trunk/patchprocess/patchReleaseAuditProblems.txt> Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5473/console> This message is automatically generated.
69. Filed HDFS-5527 to fix TestUnderReplicatedBlocks (already have a quick fix patch). The failure of TestOfflineEditsViewer is due to missing of new binary file of editsStored.
70. {color:red}-1 overall{color}. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12614660/h2832\\_20131119.patch](http://issues.apache.org/jira/secure/attachment/12614660/h2832_20131119.patch) against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 45 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:red}-1 release audit{color}. The applied patch generated 1 release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-hdfs-project/hadoop-hdfs: org.apache.hadoop.hdfs.server.namenode.ha.TestHASafeMode org.apache.hadoop.hdfs.server.blockmanagement.TestUnderReplicatedBlocks org.apache.hadoop.hdfs.tools.offlineEditsViewer.TestOfflineEditsViewer {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HDFS-Build/5488/testReport/> Release audit warnings: <https://builds.apache.org/job/PreCommit-HDFS-Build/5488/artifact/trunk/patchprocess/patchReleaseAuditProblems.txt> Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5488/console> This message is automatically generated.
71. {color:red}-1 overall{color}. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12614794/h2832\\_20131119b.patch](http://issues.apache.org/jira/secure/attachment/12614794/h2832_20131119b.patch) against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 45 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-hdfs-project/hadoop-hdfs: org.apache.hadoop.hdfs.tools.offlineEditsViewer.TestOfflineEditsViewer {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HDFS-Build/5499/testReport/> Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5499/console> This message is automatically generated.
72. {color:red}-1 overall{color}. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12615196/h2832\\_20131121.patch](http://issues.apache.org/jira/secure/attachment/12615196/h2832_20131121.patch) against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 45 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:red}-1 javadoc{color}. The javadoc tool appears to have generated -2 warning messages. {color:red}-1 eclipse:eclipse{color}. The patch failed to build with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-hdfs-project/hadoop-hdfs: org.apache.hadoop.hdfs.tools.offlineEditsViewer.TestOfflineEditsViewer {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HDFS-Build/5532/testReport/> Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5532/console> This message is automatically generated.
73. {color:red}-1 overall{color}. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12615385/h2832\\_20131122.patch](http://issues.apache.org/jira/secure/attachment/12615385/h2832_20131122.patch) against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests

- included{color}. The patch appears to include 47 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:red}-1 javadoc{color}. The javadoc tool appears to have generated -12 warning messages. {color:red}-1 eclipse:eclipse{color}. The patch failed to build with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:red}-1 release audit{color}. The applied patch generated 1 release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-hdfs-project/hadoop-hdfs: org.apache.hadoop.hdfs.tools.offlineEditsViewer.TestOfflineEditsViewer org.apache.hadoop.hdfs.TestDatanodeConfig {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HDFS-Build/5544/testReport/> Release audit warnings: <https://builds.apache.org/job/PreCommit-HDFS-Build/5544/artifact/trunk/patchprocess/patchReleaseAuditProblems.txt> Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5544/console> This message is automatically generated.
74. **body:** h2832\_20131122b.patch: 2832 branch + HDFS-5559. Note that the \-1 javadoc warnings in the previous build are nonsense. It said that there were \*-12\* warnings, i.e. 12 less warnings than trunk.  
**label:** code-design
75. {color:red}-1 overall{color}. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12615442/h2832\\_20131122b.patch](http://issues.apache.org/jira/secure/attachment/12615442/h2832_20131122b.patch) against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 49 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:red}-1 release audit{color}. The applied patch generated 1 release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-common-project/hadoop-common hadoop-hdfs-project/hadoop-hdfs hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core: org.apache.hadoop.hdfs.server.datanode.TestDeleteBlockPool org.apache.hadoop.hdfs.server.namenode.ha.TestPipelinesFailover org.apache.hadoop.hdfs.TestDatanodeConfig org.apache.hadoop.hdfs.tools.offlineEditsViewer.TestOfflineEditsViewer {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HDFS-Build/5553/testReport/> Release audit warnings: <https://builds.apache.org/job/PreCommit-HDFS-Build/5553/artifact/trunk/patchprocess/patchReleaseAuditProblems.txt> Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5553/console> This message is automatically generated.
76. The previous patch was not generated correctly -- it did not has HDFS-5559. h2832\_20131123.patch
77. {color:red}-1 overall{color}. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12615471/h2832\\_20131123.patch](http://issues.apache.org/jira/secure/attachment/12615471/h2832_20131123.patch) against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 50 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:red}-1 release audit{color}. The applied patch generated 1 release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-common-project/hadoop-common hadoop-hdfs-project/hadoop-hdfs hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core: org.apache.hadoop.hdfs.server.namenode.TestCheckpoint org.apache.hadoop.hdfs.tools.offlineEditsViewer.TestOfflineEditsViewer {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HDFS-Build/5556/testReport/> Release audit warnings: <https://builds.apache.org/job/PreCommit-HDFS-Build/5556/artifact/trunk/patchprocess/patchReleaseAuditProblems.txt> Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5556/console> This message is automatically generated.
78. {color:red}-1 overall{color}. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12615521/h2832\\_20131124.patch](http://issues.apache.org/jira/secure/attachment/12615521/h2832_20131124.patch) against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 48 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-hdfs-project/hadoop-hdfs: org.apache.hadoop.hdfs.tools.offlineEditsViewer.TestOfflineEditsViewer {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HDFS-Build/5558/testReport/> Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5558/console> This message is automatically generated.

79. Updated design doc and test plan. The test plan covers phase 1 of the implementation (DataNode is a collection of storages).
80. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12615762/20131125-HeterogeneousStorage-TestPlan.pdf> against trunk revision . {color:red}-1 patch{color}. The patch command could not apply the patch. Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5572//console> This message is automatically generated.
81. Updated patch and editsStored after merging latest changes from trunk.
82. {color:red}-1 overall{color}. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12616582/h2832\\_20131202.patch](http://issues.apache.org/jira/secure/attachment/12616582/h2832_20131202.patch) against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 48 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-hdfs-project/hadoop-hdfs: org.apache.hadoop.hdfs.tools.offlineEditsViewer.TestOfflineEditsViewer {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HDFS-Build/5608//testReport/> Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5608//console> This message is automatically generated.
83. The {{TestOfflineEditsViewer}} failure is expected and can be fixed with the attached editsStored binary file.
84. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12616642/20131202-HeterogeneousStorage-TestPlan.pdf> against trunk revision . {color:red}-1 patch{color}. The patch command could not apply the patch. Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5618//console> This message is automatically generated.
85. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12616810/20131203-HeterogeneousStorage-TestPlan.pdf> against trunk revision . {color:red}-1 patch{color}. The patch command could not apply the patch. Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5623//console> This message is automatically generated.
86. Updated merge patch to resolve recent conflicts in trunk.
87. {color:red}-1 overall{color}. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12616817/h2832\\_20131203.patch](http://issues.apache.org/jira/secure/attachment/12616817/h2832_20131203.patch) against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 48 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-hdfs-project/hadoop-hdfs: org.apache.hadoop.hdfs.tools.offlineEditsViewer.TestOfflineEditsViewer {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HDFS-Build/5624//testReport/> Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5624//console> This message is automatically generated.
88. **body:** Hi folks, sorry for just now looping back to this JIRA. I read the updated design doc, and had a few more questions: \* Storage is not always truly hierarchical, it depends on your provisioning. The current strategy of always falling back from SSD to HDD is more ambiguous when you have more than two storage types, especially with something like a tape or NAS tier. Maybe this should be configurable somehow. \* I'd like to see more discussion in the doc of migrating blocks that are currently open for short-circuit read. SCR is very common, and if this isn't handled, it makes it hard to use for an application like HBase which tries to opens all of its files once via SCR. FWIW, the HBase committers I've talked to are very interested in HSM and SSDs, so it might be helpful to get their thoughts on this topic and the feature more generally. \* Is this going to work with rolling upgrades? \* Do you foresee heartbeats and block reports always being combined in realistic scenarios? Or are there reasons to split it? Is there any additional overhead from splitting? Can we save any complexity by not supporting split reports? I see this on the test matrix. \* Have you looked at the additional memory overhead on the NN and DN from splitting up storages? With 10 disks on a DN, this could mean effectively 10x the number of DNs as before. I think this is still insignificant, but you all know better than me. \* I'd like to see more description of the client API, namely the file attribute APIs. I'll also note that LocatedBlock is not a public API; you can hack around by downcasting BlockLocation to HdfsBlockLocation to fish out the LocatedBlock, but ultimately we probably want to expose StorageType in BlockLocation itself. API examples would be great, from both the command line and the programmatic API. \* Have you put any thought about metrics and tooling to help users and admins debug their quota usage and issues with migrating files to certain storage types? Especially because of SCR. \* One of the mentioned potential uses is to do automatic migration between storage

types based on usage patterns. In this kind of scenario, it's necessary to support more expressive forms of resource management, e.g. YARN's fair scheduler. Quotas by themselves aren't sufficient. \* I think this earlier question/answer didn't make it into the doc: what happens when this file is distcp'd or copied? Arpit's earlier answer of clearing this field makes sense (or maybe we need a `{{cp -a}}` command).

**label:** code-design

89. bq. Hi folks, sorry for just now looping back to this JIRA. Is this not too late to loop back now, after design published and work started many months ago? Doing this after the merge vote is called (with 3 days to wrap up the voting) seems like a strange choice of timing to me. As regards to client APIs, we can certainly discuss them post phase 1 merge and when the work starts on phase 2 in relevant jiras. Hopefully [~arpitagarwal] can provide answers to the technical questions.
90. Most of these questions pertain to "phase 2" features, not "phase 1". If phase 1 is done and phase 2 starting, it seems now is the appropriate time to be asking phase 2 design questions. I don't have any technical issues with the code in the branch right now. I'd really appreciate if phase 1 and phase 2 (and phase x?) features could be divided up in the design doc though, since I don't think this phased implementation plan is mentioned in there right now. I'm sure it'd help other reviewers too.
91. {quote} Is there any additional overhead from splitting? {quote} DNs are not splitting block reports right now and there is no extra overhead in NN to handle it. {quote} Have you looked at the additional memory overhead on the NN and DN from splitting up storages? With 10 disks on a DN, this could mean effectively 10x the number of DNs as before. I think this is still insignificant, but you all know better than me. {quote} Yes, the bulk of the space is consumed by the block information, the volumes themselves are insignificant. {quote} I'd like to see more description of the client API, namely the file attribute APIs. I'll also note that LocatedBlock is not a public API; you can hack around by downcasting BlockLocation to HdfsBlockLocation to fish out the LocatedBlock, but ultimately we probably want to expose StorageType in BlockLocation itself. API examples would be great, from both the command line and the programmatic API. I think this earlier question/answer didn't make it into the doc: what happens when this file is distcp'd or copied? Arpit's earlier answer of clearing this field makes sense (or maybe we need a `cp -a` command). {quote} As mentioned earlier we'll document these in more detail once we start work on them i.e. post phase-1 merge. {quote} I'd like to see more discussion in the doc of migrating blocks that are currently open for short-circuit read. SCR is very common {quote} It's in the doc. The operating system, be it Unix or Windows will not let you remove a file as long as any process has a handle open to it. Even if the file looks deleted, it will remain on disk at least until the last open handle goes away. Quota permitting, the application can create additional replicas on alternate storage media while it keeps the existing handle open. {quote} Is this going to work with rolling upgrades? {quote} HDFS does not support rolling upgrades today. {quote} Storage is not always truly hierarchical, it depends on your provisioning. The current strategy of always falling back from SSD to HDD is more ambiguous when you have more than two storage types, especially with something like a tape or NAS tier. Maybe this should be configurable somehow. One of the mentioned potential uses is to do automatic migration between storage types based on usage patterns. In this kind of scenario, it's necessary to support more expressive forms of resource management, e.g. YARN's fair scheduler. Quotas by themselves aren't sufficient. {quote} We have tried to avoid the term 'hierarchical' because we are not adding any awareness of storage hierarchy. HDD is just the most sensible default although as we mention in the future section we can look into adding support for tapes etc. Automatic migration is mentioned for completeness but we are not thinking about the design for it now. Anyone from the community is free to post ideas however. Andrew, the design is the same you reviewed in August and we have discussed some of these earlier so I encourage you to read through the comment history.
92. I'm doing my best here to make this a friendly technical discussion. I guess my timing here is unfortunate with the merge vote pending, but since I don't intend to vote -1, this is just me reading the design doc again and posting more comments. Please take it as such. I'll restate that I think everything in the branch so far looks great. Arpit, I actually reviewed the previous comment history very carefully before posting, as well as the updated design document. Yes, I know that I made some of these comments before, but I brought them up in the first place because I felt like they were interesting problems related to this feature. I bring them up again because, 4 months later, I was wondering if you had any thoughts on potential solutions that could be added to the doc. It's fine if automatic migration, open files, more elaborate resource management, and additional storage types are all not in immediate scope, but I assume we'll want them in the future. I'd also like if the design doc was reworked to reflect what's in phase 1 vs. phase 2 vs. future work. This would also save me from making comments I apparently shouldn't be making on WIP sections (e.g. 6.2, 6.4). I'll say this again too, if phase 1 is done and phase 2 is starting, it seems like the design of phase 2 is exactly what be the focus of our attention right now. The back and forth: I mention SCR over and over again because HBase is very interested in using SSDs, and I figured supporting one of our biggest downstream projects would be a prime use case and potentially in scope. I'd be sad if not, but it'd be good to at least gather their requirements and see how we might get there. bq. HDFS does not support rolling upgrades today. Well, CDH supports rolling upgrades in some situations. ATM is working on metadata upgrade with HA enabled (HDFS-5138) and I've seen some recent JIRAs related to rolling upgrade (HDFS-5535), so it seems like a reasonable question. At least at the protobuf level, everything so far looks compatible, so I thought it might work as long as the handler code is compatible too. These question were



also not answered: bq. Do you foresee heartbeats and block reports always being combined in realistic scenarios? Or are there reasons to split it? Is there any additional overhead from splitting? Can we save any complexity by not supporting split reports? I see this on the test matrix. b1. Have you put any thought about metrics and tooling to help users and admins debug their quota usage and issues with migrating files to certain storage types? Especially because of SCR. Thanks Arpit. Again, this feedback isn't really merge related, it's just technical discussion. Not trying to block anything here.

93. **body:** {quote} I bring them up again because, 4 months later, I was wondering if you had any thoughts on potential solutions that could be added to the doc. It's fine if automatic migration, open files, more elaborate resource management, and additional storage types are all not in immediate scope, but I assume we'll want them in the future. {quote} Andrew, automatic migration is not in scope for our design. wrt open files can you describe a specific use case you think we should be handling that we have not described? Maybe it will help me understand your concern better. If you are concerned about reclaiming capacity for in use blocks, that is analogous to asking "If a process keeps a long-lived handle to a file what will the operating system do to reclaim disk space used by the file?" and the answer is the same - nothing. I don't want anyone reading your comments to get a false impression that the feature is incompatible with SCR. {quote} Well, CDH supports rolling upgrades in some situations. ATM is working on metadata upgrade with HA enabled (HDFS-5138) and I've seen some recent JIRAs related to rolling upgrade (HDFS-5535), so it seems like a reasonable question. At least at the protobuf level, everything so far looks compatible, so I thought it might work as long as the handler code is compatible too. {quote} I am not familiar with how CDH does rolling upgrades so I cannot tell you whether it will work. You recently bumped the layout version for caching so you might recall that HDFS layout version checks prevent a DN registering with an NN with a mismatched version. To my knowledge HDFS-5535 will not fix this limitation either. That said, we have retained wire-protocol compatibility. {quote} Do you foresee heartbeats and block reports always being combined in realistic scenarios? Or are there reasons to split it? Is there any additional overhead from splitting? Can we save any complexity by not supporting split reports? I see this on the test matrix. {quote} I thought I answered it, maybe if you describe your concerns I can give you a better answer. When the test plan says 'split' I meant splitting the reports across multiple requests. Reports will always be split by storage but we are not splitting them across multiple messages for now. What kind of overhead are you thinking of? {quote} b1. Have you put any thought about metrics and tooling to help users and admins debug their quota usage and issues with migrating files to certain storage types? {quote} We'll include it in the next design rev as we start phase 2.

**label:** code-design

94. {color:red}-1 overall{color}. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12618118/h2832\\_20131210.patch](http://issues.apache.org/jira/secure/attachment/12618118/h2832_20131210.patch) against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 48 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-hdfs-project/hadoop-hdfs: org.apache.hadoop.hdfs.server.namenode.TestListCorruptFileBlocks org.apache.hadoop.hdfs.TestDFSUpgradeFromImage org.apache.hadoop.hdfs.server.datanode.TestBlockReplacement org.apache.hadoop.hdfs.tools.offlineEditsViewer.TestOfflineEditsViewer org.apache.hadoop.hdfs.server.blockmanagement.TestNodeCount org.apache.hadoop.hdfs.server.namenode.metrics.TestNameNodeMetrics The following test timeouts occurred in hadoop-hdfs-project/hadoop-hdfs: org.apache.hadoop.hdfs.server.namenode.TestFsync org.apache.hadoop.hdfs.server.blockmanagement.TestOverReplicatedBlocks {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HDFS-Build/5690/testReport/> Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5690/console> This message is automatically generated.
95. {color:red}-1 overall{color}. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12618314/h2832\\_20131211.patch](http://issues.apache.org/jira/secure/attachment/12618314/h2832_20131211.patch) against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 48 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:red}-1 javadoc{color}. The javadoc tool appears to have generated -12 warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-hdfs-project/hadoop-hdfs: org.apache.hadoop.hdfs.tools.offlineEditsViewer.TestOfflineEditsViewer org.apache.hadoop.hdfs.server.balancer.TestBalancerWithNodeGroup

- org.apache.hadoop.hdfs.server.balancer.TestBalancerWithEncryptedTransfer {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HDFS-Build/5699/testReport/> Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5699/console> This message is automatically generated.
96. {color:red}-1 overall{color}. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12618347/h2832\\_20131211b.patch](http://issues.apache.org/jira/secure/attachment/12618347/h2832_20131211b.patch) against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 48 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-hdfs-project/hadoop-hdfs: org.apache.hadoop.hdfs.tools.offlineEditsViewer.TestOfflineEditsViewer {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HDFS-Build/5701/testReport/> Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5701/console> This message is automatically generated.
97. SUCCESS: Integrated in Hadoop-trunk-Commit #4871 (See [<https://builds.apache.org/job/Hadoop-trunk-Commit/4871/>]) svn merge --reintegrate <https://svn.apache.org/repos/asf/hadoop/common/branches/HDFS-2832> for merging Heterogeneous Storage feature branch (arp: <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1550363>) \* /hadoop/common/trunk \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/CHANGES.txt \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/docs \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/core \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSInputStream.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSOutputStream.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSUtil.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/StorageType.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/BlockListAsLongs.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/ClientProtocol.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/DatanodeID.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/DatanodeInfo.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/LayoutVersion.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/LocatedBlock.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/UnregisteredNodeException.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/ClientNamenodeProtocolServerSideTranslatorPB.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/ClientNamenodeProtocolTranslatorPB.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/DatanodeProtocolClientSideTranslatorPB.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/DatanodeProtocolServerSideTranslatorPB.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/InterDatanodeProtocolServerSideTranslatorPB.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/InterDatanodeProtocolTranslatorPB.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/PBHelper.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/balancer/Balancer.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockCollection.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockInfo.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-

hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockInfoUnderConstruction.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockManager.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockPlacementPolicy.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockPlacementPolicyDefault.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockPlacementPolicyWithNodeGroup.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlocksMap.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeDescriptor.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeManager.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeStorageInfo.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/HeartbeatManager.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/InvalidateBlocks.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/MutableBlockCollection.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/PendingDataNodeMessages.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/common/Storage.java \* /hadoop/common/trunk/hadoop-hdfs-  
project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BPOfferService.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BPServiceActor.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BlockPoolSliceScanner.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BlockReceiver.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/DataNode.java \* /hadoop/common/trunk/hadoop-  
hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/DataStorage.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/DataXceiver.java \* /hadoop/common/trunk/hadoop-  
hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/DirectoryScanner.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/Replica.java \* /hadoop/common/trunk/hadoop-  
hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/ReplicaInfo.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/StorageLocation.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/FsDatasetSpi.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/FsVolumeSpi.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsDatasetAsyncDiskService.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsDatasetImpl.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsVolumeImpl.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsVolumeList.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/ReplicaMap.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/FSDirectory.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-

hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/FSNamesystem.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/INodeFile.java \* /hadoop/common/trunk/hadoop-  
hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/NameNodeRpcServer.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/NamenodeJspHelper.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/web/resources/NamenodeWebHdfsMethods.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/BlockCommand.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/BlocksWithLocations.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/DatanodeRegistration.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/DatanodeStorage.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/StorageReport.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/tools/offlineImageViewer/ImageLoaderCurrent.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/web/JsonUtil.java \* /hadoop/common/trunk/hadoop-hdfs-  
project/hadoop-hdfs/src/main/native \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/proto/ClientNamenodeProtocol.proto \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/proto/DatanodeProtocol.proto \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/proto/InterDatanodeProtocol.proto \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/proto/hdfs.proto \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/webapps/datanode \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/webapps/hdfs \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/webapps/secondary \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/hdfs \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/DFSTestUtil.java \* /hadoop/common/trunk/hadoop-hdfs-  
project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/MiniDFSCluster.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/TestDFSShell.java \* /hadoop/common/trunk/hadoop-hdfs-  
project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/TestDFSStartupVersions.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/TestDatanodeConfig.java \* /hadoop/common/trunk/hadoop-hdfs-  
project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/TestDatanodeRegistration.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/TestFileCorruption.java \* /hadoop/common/trunk/hadoop-hdfs-  
project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/TestInjectionForSimulatedStorage.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/TestPeerCache.java \* /hadoop/common/trunk/hadoop-hdfs-  
project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/UpgradeUtilities.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/protocolPB/TestPBHelper.java \* /hadoop/common/trunk/hadoop-  
hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/balancer/TestBalancer.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockManagerTestUtil.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestBlockInfo.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestBlockInfoUnderConstruction.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestBlockManager.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestDatanodeDescriptor.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestDatanodeManager.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestHeartbeatHandling.java \*

```

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestNodeCount.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestOverReplicatedBlocks.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestPendingDataNodeMessages.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestPendingReplication.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicy.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicyWithNodeGroup.java
* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestUnderReplicatedBlocks.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/common/TestJspHelper.java * /hadoop/common/trunk/hadoop-
hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/SimulatedFSDataset.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBPOfferService.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBlockRecovery.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBlockReplacement.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBlockReport.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestDataDirs.java * /hadoop/common/trunk/hadoop-
hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestDataNodeMultipleRegistrations.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestDataNodeVolumeFailure.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestDirectoryScanner.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestSimulatedFSDataset.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/NNThroughputBenchmark.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/NameNodeAdapter.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestAddBlockRetry.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestCommitBlockSynchronization.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestDeadDatanode.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestListCorruptFileBlocks.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestNamenodeRetryCache.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/ha/TestPipelinesFailover.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/ha/TestRetryCacheWithHA.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/metrics/TestNameNodeMetrics.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/resources/editsStored.xml *
/hadoop/common/trunk/hadoop-mapreduce-project * /hadoop/common/trunk/hadoop-mapreduce-
project/CHANGES.txt * /hadoop/common/trunk/hadoop-mapreduce-project/conf *
/hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-
core/src/main/resources/mapred-default.xml

```

98. Merged the HDFS-2832 feature branch to trunk. Thanks Nicholas, Suresh, Junping and Eric!

99. SUCCESS: Integrated in Hadoop-trunk-Commit #4872 (See [<https://builds.apache.org/job/Hadoop-trunk-Commit/4872/>]) HDFS-2832: Update binary file editsStored for TestOfflineEditsViewer (arp: <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1550364>) \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/resources/editsStored
100. FAILURE: Integrated in Hadoop-Hdfs-trunk #1610 (See [<https://builds.apache.org/job/Hadoop-Hdfs-trunk/1610/>]) HDFS-2832: Update binary file editsStored for TestOfflineEditsViewer (arp: <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1550364>) \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/resources/editsStored svn merge --reintegrate <https://svn.apache.org/repos/asf/hadoop/common/branches/HDFS-2832> for merging Heterogeneous Storage feature branch (arp: <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1550363>) \*  
/hadoop/common/trunk \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/CHANGES.txt \*  
/hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/docs \*  
/hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java \*  
/hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/core \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSInputStream.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSOutputStream.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSUtil.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/StorageType.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/BlockListAsLongs.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/ClientProtocol.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/DatanodeID.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/DatanodeInfo.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/LayoutVersion.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/LocatedBlock.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/UnregisteredNodeException.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/ClientNamenodeProtocolServerSideTranslatorPB.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/ClientNamenodeProtocolTranslatorPB.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/DatanodeProtocolClientSideTranslatorPB.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/DatanodeProtocolServerSideTranslatorPB.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/InterDatanodeProtocolServerSideTranslatorPB.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/InterDatanodeProtocolTranslatorPB.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/PBHelper.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/balancer/Balancer.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockCollection.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockInfo.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockInfoUnderConstruction.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockManager.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockPlacementPolicy.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockPlacementPolicyDefault.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockPlacementPolicyWithNodeGroup.java \*  
\* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-

hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlocksMap.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeDescriptor.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeManager.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeStorageInfo.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/HeartbeatManager.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/InvalidateBlocks.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/MutableBlockCollection.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/PendingDataNodeMessages.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/common/Storage.java \* /hadoop/common/trunk/hadoop-hdfs-  
project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BPOfferService.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BPServiceActor.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BlockPoolSliceScanner.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BlockReceiver.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/DataNode.java \* /hadoop/common/trunk/hadoop-  
hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/DataStorage.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/DataXceiver.java \* /hadoop/common/trunk/hadoop-  
hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/DirectoryScanner.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/Replica.java \* /hadoop/common/trunk/hadoop-  
hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/ReplicaInfo.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/StorageLocation.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/FsDatasetSpi.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/FsVolumeSpi.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsDatasetAsyncDiskService.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsDatasetImpl.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsVolumeImpl.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsVolumeList.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/ReplicaMap.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/FSDirectory.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/FSNamesystem.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/INodeFile.java \* /hadoop/common/trunk/hadoop-  
hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/NameNodeRpcServer.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/NamenodeJspHelper.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/web/resources/NamenodeWebHdfsMethods.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/BlockCommand.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/BlocksWithLocations.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/DatanodeRegistration.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/DatanodeStorage.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/StorageReport.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/tools/offlineImageViewer/ImageLoaderCurrent.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/web/JsonUtil.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/native \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/proto/ClientNamenodeProtocol.proto \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/proto/DatanodeProtocol.proto \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/proto/InterDatanodeProtocol.proto \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/proto/hdfs.proto \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/datanode \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/hdfs \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/secondary \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/hdfs \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/DFSTestUtil.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/MiniDFSCluster.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/TestDFSShell.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/TestDFSStartupVersions.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/TestDatanodeConfig.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/TestDatanodeRegistration.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/TestFileCorruption.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/TestInjectionForSimulatedStorage.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/TestPeerCache.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/UpgradeUtilities.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/protocolPB/TestPBHelper.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/balancer/TestBalancer.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockManagerTestUtil.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestBlockInfo.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestBlockInfoUnderConstruction.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestBlockManager.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestDatanodeDescriptor.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestDatanodeManager.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestHeartbeatHandling.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestNodeCount.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestOverReplicatedBlocks.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestPendingDataNodeMessages.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestPendingReplication.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicy.java \*



```

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicyWithNodeGroup.java
* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestUnderReplicatedBlocks.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/common/TestJspHelper.java * /hadoop/common/trunk/hadoop-
hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/SimulatedFSDataset.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBPOfferService.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBlockRecovery.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBlockReplacement.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBlockReport.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestDataDirs.java * /hadoop/common/trunk/hadoop-
hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestDataNodeMultipleRegistrations.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestDataNodeVolumeFailure.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestDirectoryScanner.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestSimulatedFSDataset.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/NNThroughputBenchmark.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/NameNodeAdapter.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestAddBlockRetry.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestCommitBlockSynchronization.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestDeadDatanode.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestListCorruptFileBlocks.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestNamenodeRetryCache.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/ha/TestPipelinesFailover.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/ha/TestRetryCacheWithHA.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/metrics/TestNameNodeMetrics.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/resources/editsStored.xml *
/hadoop/common/trunk/hadoop-mapreduce-project * /hadoop/common/trunk/hadoop-mapreduce-
project/CHANGES.txt * /hadoop/common/trunk/hadoop-mapreduce-project/conf *
/hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-
core/src/main/resources/mapred-default.xml

```

101. Thanks for this significant contribution [~arpitagarwal] and [~szetszwo]!

102. FAILURE: Integrated in Hadoop-Yarn-trunk #420 (See [<https://builds.apache.org/job/Hadoop-Yarn-trunk/420/>])  
HDFS-2832: Update binary file editsStored for TestOfflineEditsViewer (arp:  
<http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1550364>) \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/resources/editsStored svn merge --reintegrate  
<https://svn.apache.org/repos/asf/hadoop/common/branches/HDFS-2832> for merging Heterogeneous Storage  
feature branch (arp: <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1550363>) \*  
/hadoop/common/trunk \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/CHANGES.txt \*  
/hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/docs \*  
/hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java \*  
/hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/core \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs \* /hadoop/common/trunk/hadoop-hdfs-

project/hadoop-hdfs/CHANGES.txt \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSInputStream.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSOutputStream.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSUtil.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/StorageType.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/BlockListAsLongs.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/ClientProtocol.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/DatanodeID.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/DatanodeInfo.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/LayoutVersion.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/LocatedBlock.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/UnregisteredNodeException.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/ClientNamenodeProtocolServerSideTranslatorPB.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/ClientNamenodeProtocolTranslatorPB.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/DatanodeProtocolClientSideTranslatorPB.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/DatanodeProtocolServerSideTranslatorPB.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/InterDatanodeProtocolServerSideTranslatorPB.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/InterDatanodeProtocolTranslatorPB.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/PBHelper.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/balancer/Balancer.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockCollection.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockInfo.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockInfoUnderConstruction.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockManager.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockPlacementPolicy.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockPlacementPolicyDefault.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockPlacementPolicyWithNodeGroup.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlocksMap.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeDescriptor.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeManager.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeStorageInfo.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/HeartbeatManager.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/InvalidateBlocks.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/MutableBlockCollection.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/PendingDataNodeMessages.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/common/Storage.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BPOfferService.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BPServiceActor.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BlockPoolSliceScanner.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BlockReceiver.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/DataNode.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/DataStorage.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/DataXceiver.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/DirectoryScanner.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/Replica.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/ReplicaInfo.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/StorageLocation.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/FsDatasetSpi.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/FsVolumeSpi.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsDatasetAsyncDiskService.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsDatasetImpl.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsVolumeImpl.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsVolumeList.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/ReplicaMap.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/FSDirectory.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/FSNamesystem.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/INodeFile.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/NameNodeRpcServer.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/NamenodeJspHelper.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/web/resources/NamenodeWebHdfsMethods.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/BlockCommand.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/BlocksWithLocations.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/DatanodeRegistration.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/DatanodeStorage.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/StorageReport.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/tools/offlineImageViewer/ImageLoaderCurrent.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/web/JsonUtil.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/native \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/proto/ClientNamenodeProtocol.proto \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/proto/DatanodeProtocol.proto \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-

hdfs/src/main/proto/InterDatanodeProtocol.proto \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/proto/hdfs.proto \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/datanode \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/hdfs \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/secondary \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/hdfs \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/DFSTestUtil.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/MiniDFSCluster.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/TestDFSShell.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/TestDFSStartupVersions.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/TestDatanodeConfig.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/TestDatanodeRegistration.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/TestFileCorruption.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/TestInjectionForSimulatedStorage.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/TestPeerCache.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/UpgradeUtilities.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/protocolPB/TestPBHelper.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/balancer/TestBalancer.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockManagerTestUtil.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestBlockInfo.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestBlockInfoUnderConstruction.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestBlockManager.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestDatanodeDescriptor.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestDatanodeManager.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestHeartbeatHandling.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestNodeCount.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestOverReplicatedBlocks.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestPendingDataNodeMessages.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestPendingReplication.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicy.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicyWithNodeGroup.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestUnderReplicatedBlocks.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/common/TestJspHelper.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/SimulatedFSDataset.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBPOfferService.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBlockRecovery.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBlockReplacement.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBlockReport.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestDataDirs.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestDataNodeMultipleRegistrations.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestDataNodeVolumeFailure.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestDirectoryScanner.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestSimulatedFSDataset.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/NNThroughputBenchmark.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/NameNodeAdapter.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestAddBlockRetry.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestCommitBlockSynchronization.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestDeadDatanode.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestListCorruptFileBlocks.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestNamenodeRetryCache.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/ha/TestPipelinesFailover.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/ha/TestRetryCacheWithHA.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/metrics/TestNameNodeMetrics.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/resources/editsStored.xml \* /hadoop/common/trunk/hadoop-mapreduce-project \* /hadoop/common/trunk/hadoop-mapreduce-project/CHANGES.txt \* /hadoop/common/trunk/hadoop-mapreduce-project/conf \* /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client-core/src/main/resources/mapred-default.xml

103. FAILURE: Integrated in Hadoop-Mapreduce-trunk #1637 (See [<https://builds.apache.org/job/Hadoop-Mapreduce-trunk/1637/>]) HDFS-2832: Update binary file editsStored for TestOfflineEditsViewer (arp: <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1550364>) \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/resources/editsStored svn merge --reintegrate <https://svn.apache.org/repos/asf/hadoop/common/branches/HDFS-2832> for merging Heterogeneous Storage feature branch (arp: <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1550363>) \* /hadoop/common/trunk \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/CHANGES.txt \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/docs \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/core \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSInputStream.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSOutputStream.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSUtil.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/StorageType.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/BlockListAsLongs.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/ClientProtocol.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/DatanodeID.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/DatanodeInfo.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/LayoutVersion.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/LocatedBlock.java \* /hadoop/common/trunk/hadoop-hdfs-

project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocol/UnregisteredNodeException.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/ClientNamenodeProtocolServerSideTranslatorPB.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/ClientNamenodeProtocolTranslatorPB.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/DatanodeProtocolClientSideTranslatorPB.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/DatanodeProtocolServerSideTranslatorPB.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/InterDatanodeProtocolServerSideTranslatorPB.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/InterDatanodeProtocolTranslatorPB.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/protocolPB/PBHelper.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/balancer/Balancer.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockCollection.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockInfo.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockInfoUnderConstruction.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockManager.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockPlacementPolicy.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockPlacementPolicyDefault.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockPlacementPolicyWithNodeGroup.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/BlocksMap.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeDescriptor.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeManager.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeStorageInfo.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/HeartbeatManager.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/InvalidateBlocks.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/MutableBlockCollection.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/PendingDataNodeMessages.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/common/Storage.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BPOfferService.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BPServiceActor.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BlockPoolSliceScanner.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/BlockReceiver.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/DataNode.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/DataStorage.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/DataXceiver.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/DirectoryScanner.java \*

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-

hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/Replica.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/ReplicaInfo.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/StorageLocation.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/FsDatasetSpi.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/FsVolumeSpi.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsDatasetAsyncDiskService.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsDatasetImpl.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsVolumeImpl.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/FsVolumeList.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/fsdataset/impl/ReplicaMap.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/FSDirectory.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/FSNamesystem.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/INodeFile.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/NameNodeRpcServer.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/NamenodeJspHelper.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode/web/resources/NamenodeWebHdfsMethods.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/BlockCommand.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/BlocksWithLocations.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/DatanodeRegistration.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/DatanodeStorage.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/protocol/StorageReport.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/tools/offlineImageViewer/ImageLoaderCurrent.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/web/JsonUtil.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/native \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/proto/ClientNamenodeProtocol.proto \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/proto/DatanodeProtocol.proto \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/proto/InterDatanodeProtocol.proto \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/proto/hdfs.proto \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/datanode \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/hdfs \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/secondary \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/hdfs \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/DFSTestUtil.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/MiniDFSCluster.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/TestDFSShell.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/TestDFSStartupVersions.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/TestDatanodeConfig.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/TestDatanodeRegistration.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/TestFileCorruption.java \* /hadoop/common/trunk/hadoop-hdfs-

project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/TestInjectionForSimulatedStorage.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/TestPeerCache.java \* /hadoop/common/trunk/hadoop-hdfs-  
project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/UpgradeUtilities.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/protocolPB/TestPBHelper.java \* /hadoop/common/trunk/hadoop-  
hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/balancer/TestBalancer.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/BlockManagerTestUtil.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestBlockInfo.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestBlockInfoUnderConstruction.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestBlockManager.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestDatanodeDescriptor.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestDatanodeManager.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestHeartbeatHandling.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestNodeCount.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestOverReplicatedBlocks.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestPendingDataNodeMessages.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestPendingReplication.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicy.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestReplicationPolicyWithNodeGroup.java  
\* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/blockmanagement/TestUnderReplicatedBlocks.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/common/TestJspHelper.java \* /hadoop/common/trunk/hadoop-  
hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/SimulatedFSDataset.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBPOfferService.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBlockRecovery.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBlockReplacement.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestBlockReport.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestDataDirs.java \* /hadoop/common/trunk/hadoop-  
hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestDataNodeMultipleRegistrations.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestDataNodeVolumeFailure.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestDirectoryScanner.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/datanode/TestSimulatedFSDataset.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/NNThroughputBenchmark.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/NameNodeAdapter.java \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-  
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestAddBlockRetry.java \*



```

/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestCommitBlockSynchronization.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestDeadDatanode.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestListCorruptFileBlocks.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/TestNamenodeRetryCache.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/ha/TestPipelinesFailover.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/ha/TestRetryCacheWithHA.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode/metrics/TestNameNodeMetrics.java *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/resources/editsStored.xml *
/hadoop/common/trunk/hadoop-mapreduce-project */hadoop/common/trunk/hadoop-mapreduce-
project/CHANGES.txt */hadoop/common/trunk/hadoop-mapreduce-project/conf */
/hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-
core/src/main/resources/mapred-default.xml

```

104. Phase 2 tasks will be tracked under HDFS-5682.
105. Preliminary merge patch for branch-2, pending full test run.
106. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12620579/editsStored> against trunk revision . {color:red}-1 patch{color}. The patch command could not apply the patch. Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5795//console> This message is automatically generated.
107. Merge patch rebased to current branch-2.
108. {color:red}-1 overall{color}. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12621416/h2832\\_branch-2\\_20140103.patch](http://issues.apache.org/jira/secure/attachment/12621416/h2832_branch-2_20140103.patch) against trunk revision . {color:red}-1 patch{color}. The patch command could not apply the patch. Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/5821//console> This message is automatically generated.
109. HDFS-2832 changes have been merged from trunk to branch-2. The fix and target versions have been updated to 2.4.0.
110. SUCCESS: Integrated in Hadoop-trunk-Commit #4965 (See [<https://builds.apache.org/job/Hadoop-trunk-Commit/4965/>]) HDFS-2832. Update CHANGES.txt to reflect merge to branch-2 (arp: <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1556088>) \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt
111. SUCCESS: Integrated in Hadoop-Yarn-trunk #445 (See [<https://builds.apache.org/job/Hadoop-Yarn-trunk/445/>]) HDFS-2832. Update CHANGES.txt to reflect merge to branch-2 (arp: <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1556088>) \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt
112. FAILURE: Integrated in Hadoop-Hdfs-trunk #1637 (See [<https://builds.apache.org/job/Hadoop-Hdfs-trunk/1637/>]) HDFS-2832. Update CHANGES.txt to reflect merge to branch-2 (arp: <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1556088>) \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt
113. SUCCESS: Integrated in Hadoop-Mapreduce-trunk #1662 (See [<https://builds.apache.org/job/Hadoop-Mapreduce-trunk/1662/>]) HDFS-2832. Update CHANGES.txt to reflect merge to branch-2 (arp: <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1556088>) \*  
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt
114. Arun, I'm changing this fixversion back to 2.3.0 since the HDFS-2832 subtasks were included. Maybe HDFS-5682 (the second phase) is targeted at 2.4.0.
115. Thanks for catching this Andrew. Resolving it to avoid getting caught in future bulk edits.
116. hey guys, it seems that this feature has been included in hadoop 2.3. But how can I use it in my cluster? Where is the user document?
117. Jiahongchao, we don't expose any user API yet for HSM. This is going to be added in HDFS-5682. It might have been a mistake to advertise HSM in the release note at this stage of development, since I imagine we'll be getting a lot of questions like this.
118. Closing old tickets that are already shipped in a release.