

git_comments:

1. * * @see #getDelegationTokens(String) * This is similar to getDelegationTokens, with the added restriction that if * a token is already present in the passed Credentials object - that token * is returned instead of a new delegation token. * * If the token is found to be cached in the Credentials object, this API does * not verify the token validity or the passed in renewer. * * * @param renewer the account name that is allowed to renew the token. * @param credentials a Credentials object containing already knowing * delegationTokens. * @return a list of delegation tokens. * @throws IOException
2. FileSystem

git_commits:

1. **summary:** HADOOP-7933. Add a getDelegationTokens api to FileSystem which checks for known tokens in the passed Credentials object. (sseth)
message: HADOOP-7933. Add a getDelegationTokens api to FileSystem which checks for known tokens in the passed Credentials object. (sseth) git-svn-id: <https://svn.apache.org/repos/asf/hadoop/common/trunk@1226916> 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Viewfs changes for MAPREDUCE-3529
description: ViewFs.getDelegationTokens returns a list of tokens for the associated namenodes. Credentials serializes these tokens using the service name for the actual namenodes. Effectively, tokens are not cached for viewfs (some more details in MR 3529). Affects any job which uses the TokenCache in tasks along with viewfs (some Pig jobs). Talk to Jitendra about this, some options 1. Change Credentials.getAllTokens to return the key, instead of just a token list (associate the viewfs canonical name with a token in credentials) 2. Have viewfs issue a fake token. Both of these would allow for a single viewfs configuration only. 3. An additional API in FileSystem - something like getDelegationTokens(String renewer, Credentials credentials) - which would check the credentials object before making token requests to the actual namenode. 4. An additional API in FileSystem - getCanonicalServiceNames - similar to getDelegationTokens, which would return service names for the actual namenodes. TokenCache/Credentials can work using this list. 5. have getDelegationTokens check the current UGI - and fetch tokens only if they don't exist. Have a quick patch for 3, along with associated MR changes.
label: documentation
2. **summary:** Viewfs changes for MAPREDUCE-3529
description: ViewFs.getDelegationTokens returns a list of tokens for the associated namenodes. Credentials serializes these tokens using the service name for the actual namenodes. Effectively, tokens are not cached for viewfs (some more details in MR 3529). Affects any job which uses the TokenCache in tasks along with viewfs (some Pig jobs). Talk to Jitendra about this, some options 1. Change Credentials.getAllTokens to return the key, instead of just a token list (associate the viewfs canonical name with a token in credentials) 2. Have viewfs issue a fake token. Both of these would allow for a single viewfs configuration only. 3. An additional API in FileSystem - something like getDelegationTokens(String renewer, Credentials credentials) - which would check the credentials object before making token requests to the actual namenode. 4. An additional API in FileSystem - getCanonicalServiceNames - similar to getDelegationTokens, which would return service names for the actual namenodes. TokenCache/Credentials can work using this list. 5. have getDelegationTokens check

the current UGI - and fetch tokens only if they don't exist. Have a quick patch for 3, along with associated MR changes.

3. **summary:** Viewfs changes for MAPREDUCE-3529

description: ViewFs.getDelegationTokens returns a list of tokens for the associated namenodes. Credentials serializes these tokens using the service name for the actual namenodes. Effectively, tokens are not cached for viewfs (some more details in MR 3529). Affects any job which uses the TokenCache in tasks along with viewfs (some Pig jobs). Talk to Jitendra about this, some options 1. Change Credentials.getAllTokens to return the key, instead of just a token list (associate the viewfs canonical name with a token in credentials) 2. Have viewfs issue a fake token. Both of these would allow for a single viewfs configuration only. 3. An additional API in FileSystem - something like getDelegationTokens(String renewer, Credentials credentials) - which would check the credentials object before making token requests to the actual namenode. 4. An additional API in FileSystem - getCanonicalServiceNames - similar to getDelegationTokens, which would return service names for the actual namenodes. TokenCache/Credentials can work using this list. 5. have getDelegationTokens check the current UGI - and fetch tokens only if they don't exist. Have a quick patch for 3, along with associated MR changes.

4. **summary:** Viewfs changes for MAPREDUCE-3529

description: ViewFs.getDelegationTokens returns a list of tokens for the associated namenodes. Credentials serializes these tokens using the service name for the actual namenodes. Effectively, tokens are not cached for viewfs (some more details in MR 3529). Affects any job which uses the TokenCache in tasks along with viewfs (some Pig jobs). Talk to Jitendra about this, some options 1. Change Credentials.getAllTokens to return the key, instead of just a token list (associate the viewfs canonical name with a token in credentials) 2. Have viewfs issue a fake token. Both of these would allow for a single viewfs configuration only. 3. An additional API in FileSystem - something like getDelegationTokens(String renewer, Credentials credentials) - which would check the credentials object before making token requests to the actual namenode. 4. An additional API in FileSystem - getCanonicalServiceNames - similar to getDelegationTokens, which would return service names for the actual namenodes. TokenCache/Credentials can work using this list. 5. have getDelegationTokens check the current UGI - and fetch tokens only if they don't exist. Have a quick patch for 3, along with associated MR changes.

label: code-design

5. **summary:** Viewfs changes for MAPREDUCE-3529

description: ViewFs.getDelegationTokens returns a list of tokens for the associated namenodes. Credentials serializes these tokens using the service name for the actual namenodes. Effectively, tokens are not cached for viewfs (some more details in MR 3529). Affects any job which uses the TokenCache in tasks along with viewfs (some Pig jobs). Talk to Jitendra about this, some options 1. Change Credentials.getAllTokens to return the key, instead of just a token list (associate the viewfs canonical name with a token in credentials) 2. Have viewfs issue a fake token. Both of these would allow for a single viewfs configuration only. 3. An additional API in FileSystem - something like getDelegationTokens(String renewer, Credentials credentials) - which would check the credentials object before making token requests to the actual namenode. 4. An additional API in FileSystem - getCanonicalServiceNames - similar to getDelegationTokens, which would return service names for the actual namenodes. TokenCache/Credentials can work using this list. 5. have getDelegationTokens check the current UGI - and fetch tokens only if they don't exist. Have a quick patch for 3, along with associated MR changes.

6. **summary:** Viewfs changes for MAPREDUCE-3529

description: ViewFs.getDelegationTokens returns a list of tokens for the associated namenodes. Credentials serializes these tokens using the service name for the actual namenodes. Effectively, tokens are not cached for viewfs (some more details in MR 3529). Affects any job which uses the TokenCache in tasks along with viewfs (some Pig jobs). Talk to Jitendra about this, some options 1. Change Credentials.getAllTokens to return the key, instead of just a token list (associate the viewfs canonical name with a token in credentials) 2. Have viewfs issue a fake token. Both of these would allow for a single viewfs configuration only. 3. An additional API in FileSystem - something like getDelegationTokens(String renewer, Credentials credentials) - which would check the credentials object before making token requests to the actual namenode. 4. An additional API in FileSystem - getCanonicalServiceNames - similar to getDelegationTokens, which would return service names for the actual namenodes. TokenCache/Credentials can work using this list. 5. have getDelegationTokens check the current UGI - and fetch tokens only if they don't exist. Have a quick patch for 3, along with associated MR changes.

7. **summary:** Viewfs changes for MAPREDUCE-3529

description: ViewFs.getDelegationTokens returns a list of tokens for the associated namenodes. Credentials serializes these tokens using the service name for the actual namenodes. Effectively, tokens are not cached for viewfs (some more details in MR 3529). Affects any job which uses the TokenCache in tasks along with viewfs (some Pig jobs). Talk to Jitendra about this, some options 1. Change Credentials.getAllTokens to return the key, instead of just a token list (associate the viewfs canonical name with a token in credentials) 2. Have viewfs issue a fake token. Both of these would allow for a single viewfs configuration only. 3. An additional API in FileSystem - something like getDelegationTokens(String renewer, Credentials credentials) - which would check the credentials object before making token requests to the actual namenode. 4. An additional API in FileSystem - getCanonicalServiceNames - similar to getDelegationTokens, which would return service names for the actual namenodes. TokenCache/Credentials can work using this list. 5. have getDelegationTokens check the current UGI - and fetch tokens only if they don't exist. Have a quick patch for 3, along with associated MR changes.

8. **summary:** Viewfs changes for MAPREDUCE-3529

description: ViewFs.getDelegationTokens returns a list of tokens for the associated namenodes. Credentials serializes these tokens using the service name for the actual namenodes. Effectively, tokens are not cached for viewfs (some more details in MR 3529). Affects any job which uses the TokenCache in tasks along with viewfs (some Pig jobs). Talk to Jitendra about this, some options 1. Change Credentials.getAllTokens to return the key, instead of just a token list (associate the viewfs canonical name with a token in credentials) 2. Have viewfs issue a fake token. Both of these would allow for a single viewfs configuration only. 3. An additional API in FileSystem - something like getDelegationTokens(String renewer, Credentials credentials) - which would check the credentials object before making token requests to the actual namenode. 4. An additional API in FileSystem - getCanonicalServiceNames - similar to getDelegationTokens, which would return service names for the actual namenodes. TokenCache/Credentials can work using this list. 5. have getDelegationTokens check the current UGI - and fetch tokens only if they don't exist. Have a quick patch for 3, along with associated MR changes.

9. **summary:** Viewfs changes for MAPREDUCE-3529

description: ViewFs.getDelegationTokens returns a list of tokens for the associated namenodes. Credentials serializes these tokens using the service name for the actual namenodes. Effectively, tokens are not cached for viewfs (some more details in MR 3529). Affects any job which uses the TokenCache in tasks along with viewfs (some Pig jobs). Talk to Jitendra about this, some options 1. Change Credentials.getAllTokens to return the key, instead of just a token list (associate the viewfs canonical name with a token in credentials) 2. Have viewfs issue a fake token. Both of these would allow for a single viewfs configuration only. 3. An additional API in FileSystem - something like getDelegationTokens(String renewer, Credentials credentials) - which would check the credentials object before making token requests to the actual namenode. 4. An additional API in FileSystem - getCanonicalServiceNames - similar to getDelegationTokens, which would return service names for the actual namenodes. TokenCache/Credentials can work using this list. 5. have getDelegationTokens check the current UGI - and fetch tokens only if they don't exist. Have a quick patch for 3, along with associated MR changes.

10. **summary:** Viewfs changes for MAPREDUCE-3529

description: ViewFs.getDelegationTokens returns a list of tokens for the associated namenodes. Credentials serializes these tokens using the service name for the actual namenodes. Effectively, tokens are not cached for viewfs (some more details in MR 3529). Affects any job which uses the TokenCache in tasks along with viewfs (some Pig jobs). Talk to Jitendra about this, some options 1. Change Credentials.getAllTokens to return the key, instead of just a token list (associate the viewfs canonical name with a token in credentials) 2. Have viewfs issue a fake token. Both of these would allow for a single viewfs configuration only. 3. An additional API in FileSystem - something like getDelegationTokens(String renewer, Credentials credentials) - which would check the credentials object before making token requests to the actual namenode. 4. An additional API in FileSystem - getCanonicalServiceNames - similar to getDelegationTokens, which would return service names for the actual namenodes. TokenCache/Credentials can work using this list. 5. have getDelegationTokens check the current UGI - and fetch tokens only if they don't exist. Have a quick patch for 3, along with associated MR changes.

11. **summary:** Viewfs changes for MAPREDUCE-3529

description: ViewFs.getDelegationTokens returns a list of tokens for the associated namenodes. Credentials serializes these tokens using the service name for the actual namenodes. Effectively, tokens

are not cached for viewfs (some more details in MR 3529). Affects any job which uses the TokenCache in tasks along with viewfs (some Pig jobs). Talk to Jitendra about this, some options 1. Change Credentials.getAllTokens to return the key, instead of just a token list (associate the viewfs canonical name with a token in credentials) 2. Have viewfs issue a fake token. Both of these would allow for a single viewfs configuration only. 3. An additional API in FileSystem - something like getDelegationTokens(String renewer, Credentials credentials) - which would check the credentials object before making token requests to the actual namenode. 4. An additional API in FileSystem - getCanonicalServiceNames - similar to getDelegationTokens, which would return service names for the actual namenodes. TokenCache/Credentials can work using this list. 5. have getDelegationTokens check the current UGI - and fetch tokens only if they don't exist. Have a quick patch for 3, along with associated MR changes.

12. **summary:** Viewfs changes for MAPREDUCE-3529

description: ViewFs.getDelegationTokens returns a list of tokens for the associated namenodes. Credentials serializes these tokens using the service name for the actual namenodes. Effectively, tokens are not cached for viewfs (some more details in MR 3529). Affects any job which uses the TokenCache in tasks along with viewfs (some Pig jobs). Talk to Jitendra about this, some options 1. Change Credentials.getAllTokens to return the key, instead of just a token list (associate the viewfs canonical name with a token in credentials) 2. Have viewfs issue a fake token. Both of these would allow for a single viewfs configuration only. 3. An additional API in FileSystem - something like getDelegationTokens(String renewer, Credentials credentials) - which would check the credentials object before making token requests to the actual namenode. 4. An additional API in FileSystem - getCanonicalServiceNames - similar to getDelegationTokens, which would return service names for the actual namenodes. TokenCache/Credentials can work using this list. 5. have getDelegationTokens check the current UGI - and fetch tokens only if they don't exist. Have a quick patch for 3, along with associated MR changes.

13. **summary:** Viewfs changes for MAPREDUCE-3529

description: ViewFs.getDelegationTokens returns a list of tokens for the associated namenodes. Credentials serializes these tokens using the service name for the actual namenodes. Effectively, tokens are not cached for viewfs (some more details in MR 3529). Affects any job which uses the TokenCache in tasks along with viewfs (some Pig jobs). Talk to Jitendra about this, some options 1. Change Credentials.getAllTokens to return the key, instead of just a token list (associate the viewfs canonical name with a token in credentials) 2. Have viewfs issue a fake token. Both of these would allow for a single viewfs configuration only. 3. An additional API in FileSystem - something like getDelegationTokens(String renewer, Credentials credentials) - which would check the credentials object before making token requests to the actual namenode. 4. An additional API in FileSystem - getCanonicalServiceNames - similar to getDelegationTokens, which would return service names for the actual namenodes. TokenCache/Credentials can work using this list. 5. have getDelegationTokens check the current UGI - and fetch tokens only if they don't exist. Have a quick patch for 3, along with associated MR changes.

14. **summary:** Viewfs changes for MAPREDUCE-3529

description: ViewFs.getDelegationTokens returns a list of tokens for the associated namenodes. Credentials serializes these tokens using the service name for the actual namenodes. Effectively, tokens are not cached for viewfs (some more details in MR 3529). Affects any job which uses the TokenCache in tasks along with viewfs (some Pig jobs). Talk to Jitendra about this, some options 1. Change Credentials.getAllTokens to return the key, instead of just a token list (associate the viewfs canonical name with a token in credentials) 2. Have viewfs issue a fake token. Both of these would allow for a single viewfs configuration only. 3. An additional API in FileSystem - something like getDelegationTokens(String renewer, Credentials credentials) - which would check the credentials object before making token requests to the actual namenode. 4. An additional API in FileSystem - getCanonicalServiceNames - similar to getDelegationTokens, which would return service names for the actual namenodes. TokenCache/Credentials can work using this list. 5. have getDelegationTokens check the current UGI - and fetch tokens only if they don't exist. Have a quick patch for 3, along with associated MR changes.

15. **summary:** Viewfs changes for MAPREDUCE-3529

description: ViewFs.getDelegationTokens returns a list of tokens for the associated namenodes. Credentials serializes these tokens using the service name for the actual namenodes. Effectively, tokens are not cached for viewfs (some more details in MR 3529). Affects any job which uses the TokenCache in tasks along with viewfs (some Pig jobs). Talk to Jitendra about this, some options 1. Change Credentials.getAllTokens to return the key, instead of just a token list (associate the viewfs canonical

name with a token in credentials) 2. Have viewfs issue a fake token. Both of these would allow for a single viewfs configuration only. 3. An additional API in FileSystem - something like `getDelegationTokens(String renewer, Credentials credentials)` - which would check the credentials object before making token requests to the actual namenode. 4. An additional API in FileSystem - `getCanonicalServiceNames` - similar to `getDelegationTokens`, which would return service names for the actual namenodes. TokenCache/Credentials can work using this list. 5. have `getDelegationTokens` check the current UGI - and fetch tokens only if they don't exist. Have a quick patch for 3, along with associated MR changes.

16. **summary:** Viewfs changes for MAPREDUCE-3529

description: ViewFs.`getDelegationTokens` returns a list of tokens for the associated namenodes. Credentials serializes these tokens using the service name for the actual namenodes. Effectively, tokens are not cached for viewfs (some more details in MR 3529). Affects any job which uses the TokenCache in tasks along with viewfs (some Pig jobs). Talk to Jitendra about this, some options 1. Change `Credentials.getAllTokens` to return the key, instead of just a token list (associate the viewfs canonical name with a token in credentials) 2. Have viewfs issue a fake token. Both of these would allow for a single viewfs configuration only. 3. An additional API in FileSystem - something like `getDelegationTokens(String renewer, Credentials credentials)` - which would check the credentials object before making token requests to the actual namenode. 4. An additional API in FileSystem - `getCanonicalServiceNames` - similar to `getDelegationTokens`, which would return service names for the actual namenodes. TokenCache/Credentials can work using this list. 5. have `getDelegationTokens` check the current UGI - and fetch tokens only if they don't exist. Have a quick patch for 3, along with associated MR changes.

17. **summary:** Viewfs changes for MAPREDUCE-3529

description: ViewFs.`getDelegationTokens` returns a list of tokens for the associated namenodes. Credentials serializes these tokens using the service name for the actual namenodes. Effectively, tokens are not cached for viewfs (some more details in MR 3529). Affects any job which uses the TokenCache in tasks along with viewfs (some Pig jobs). Talk to Jitendra about this, some options 1. Change `Credentials.getAllTokens` to return the key, instead of just a token list (associate the viewfs canonical name with a token in credentials) 2. Have viewfs issue a fake token. Both of these would allow for a single viewfs configuration only. 3. An additional API in FileSystem - something like `getDelegationTokens(String renewer, Credentials credentials)` - which would check the credentials object before making token requests to the actual namenode. 4. An additional API in FileSystem - `getCanonicalServiceNames` - similar to `getDelegationTokens`, which would return service names for the actual namenodes. TokenCache/Credentials can work using this list. 5. have `getDelegationTokens` check the current UGI - and fetch tokens only if they don't exist. Have a quick patch for 3, along with associated MR changes.

18. **summary:** Viewfs changes for MAPREDUCE-3529

description: ViewFs.`getDelegationTokens` returns a list of tokens for the associated namenodes. Credentials serializes these tokens using the service name for the actual namenodes. Effectively, tokens are not cached for viewfs (some more details in MR 3529). Affects any job which uses the TokenCache in tasks along with viewfs (some Pig jobs). Talk to Jitendra about this, some options 1. Change `Credentials.getAllTokens` to return the key, instead of just a token list (associate the viewfs canonical name with a token in credentials) 2. Have viewfs issue a fake token. Both of these would allow for a single viewfs configuration only. 3. An additional API in FileSystem - something like `getDelegationTokens(String renewer, Credentials credentials)` - which would check the credentials object before making token requests to the actual namenode. 4. An additional API in FileSystem - `getCanonicalServiceNames` - similar to `getDelegationTokens`, which would return service names for the actual namenodes. TokenCache/Credentials can work using this list. 5. have `getDelegationTokens` check the current UGI - and fetch tokens only if they don't exist. Have a quick patch for 3, along with associated MR changes.

19. **summary:** Viewfs changes for MAPREDUCE-3529

description: ViewFs.`getDelegationTokens` returns a list of tokens for the associated namenodes. Credentials serializes these tokens using the service name for the actual namenodes. Effectively, tokens are not cached for viewfs (some more details in MR 3529). Affects any job which uses the TokenCache in tasks along with viewfs (some Pig jobs). Talk to Jitendra about this, some options 1. Change `Credentials.getAllTokens` to return the key, instead of just a token list (associate the viewfs canonical name with a token in credentials) 2. Have viewfs issue a fake token. Both of these would allow for a single viewfs configuration only. 3. An additional API in FileSystem - something like `getDelegationTokens(String renewer, Credentials credentials)` - which would check the credentials object

before making token requests to the actual namenode. 4. An additional API in FileSystem - getCanonicalServiceNames - similar to getDelegationTokens, which would return service names for the actual namenodes. TokenCache/Credentials can work using this list. 5. have getDelegationTokens check the current UGI - and fetch tokens only if they don't exist. Have a quick patch for 3, along with associated MR changes.

label: code-design

20. **summary:** Viewfs changes for MAPREDUCE-3529

description: ViewFs.getDelegationTokens returns a list of tokens for the associated namenodes. Credentials serializes these tokens using the service name for the actual namenodes. Effectively, tokens are not cached for viewfs (some more details in MR 3529). Affects any job which uses the TokenCache in tasks along with viewfs (some Pig jobs). Talk to Jitendra about this, some options 1. Change Credentials.getAllTokens to return the key, instead of just a token list (associate the viewfs canonical name with a token in credentials) 2. Have viewfs issue a fake token. Both of these would allow for a single viewfs configuration only. 3. An additional API in FileSystem - something like getDelegationTokens(String renewer, Credentials credentials) - which would check the credentials object before making token requests to the actual namenode. 4. An additional API in FileSystem - getCanonicalServiceNames - similar to getDelegationTokens, which would return service names for the actual namenodes. TokenCache/Credentials can work using this list. 5. have getDelegationTokens check the current UGI - and fetch tokens only if they don't exist. Have a quick patch for 3, along with associated MR changes.

21. **summary:** Viewfs changes for MAPREDUCE-3529

description: ViewFs.getDelegationTokens returns a list of tokens for the associated namenodes. Credentials serializes these tokens using the service name for the actual namenodes. Effectively, tokens are not cached for viewfs (some more details in MR 3529). Affects any job which uses the TokenCache in tasks along with viewfs (some Pig jobs). Talk to Jitendra about this, some options 1. Change Credentials.getAllTokens to return the key, instead of just a token list (associate the viewfs canonical name with a token in credentials) 2. Have viewfs issue a fake token. Both of these would allow for a single viewfs configuration only. 3. An additional API in FileSystem - something like getDelegationTokens(String renewer, Credentials credentials) - which would check the credentials object before making token requests to the actual namenode. 4. An additional API in FileSystem - getCanonicalServiceNames - similar to getDelegationTokens, which would return service names for the actual namenodes. TokenCache/Credentials can work using this list. 5. have getDelegationTokens check the current UGI - and fetch tokens only if they don't exist. Have a quick patch for 3, along with associated MR changes.

22. **summary:** Viewfs changes for MAPREDUCE-3529

description: ViewFs.getDelegationTokens returns a list of tokens for the associated namenodes. Credentials serializes these tokens using the service name for the actual namenodes. Effectively, tokens are not cached for viewfs (some more details in MR 3529). Affects any job which uses the TokenCache in tasks along with viewfs (some Pig jobs). Talk to Jitendra about this, some options 1. Change Credentials.getAllTokens to return the key, instead of just a token list (associate the viewfs canonical name with a token in credentials) 2. Have viewfs issue a fake token. Both of these would allow for a single viewfs configuration only. 3. An additional API in FileSystem - something like getDelegationTokens(String renewer, Credentials credentials) - which would check the credentials object before making token requests to the actual namenode. 4. An additional API in FileSystem - getCanonicalServiceNames - similar to getDelegationTokens, which would return service names for the actual namenodes. TokenCache/Credentials can work using this list. 5. have getDelegationTokens check the current UGI - and fetch tokens only if they don't exist. Have a quick patch for 3, along with associated MR changes.

23. **summary:** Viewfs changes for MAPREDUCE-3529

description: ViewFs.getDelegationTokens returns a list of tokens for the associated namenodes. Credentials serializes these tokens using the service name for the actual namenodes. Effectively, tokens are not cached for viewfs (some more details in MR 3529). Affects any job which uses the TokenCache in tasks along with viewfs (some Pig jobs). Talk to Jitendra about this, some options 1. Change Credentials.getAllTokens to return the key, instead of just a token list (associate the viewfs canonical name with a token in credentials) 2. Have viewfs issue a fake token. Both of these would allow for a single viewfs configuration only. 3. An additional API in FileSystem - something like getDelegationTokens(String renewer, Credentials credentials) - which would check the credentials object before making token requests to the actual namenode. 4. An additional API in FileSystem - getCanonicalServiceNames - similar to getDelegationTokens, which would return service names for the

actual namenodes. TokenCache/Credentials can work using this list. 5. have getDelegationTokens check the current UGI - and fetch tokens only if they don't exist. Have a quick patch for 3, along with associated MR changes.

jira_issues_comments:

1. **body:** I think, option 3 is ok. It is an optimization over the existing API. Please specify in the javadoc that if a token is found for the filesystem, the api returns it without verifying the expiry or renewer. Also since Credentials class is LimitedPrivate for hdfs and mapreduce, we should annotate this API also with limitedprivate.
label: documentation
2. Added a new API to FileSystem - getDelegationTokens(String , Credentials) If this change looks Ok, will make similar changes for AbstractFileSystem and FileContext. Also, the new API gets delegation tokens once per service name (instead of once per mount point). The same change can be made for getDelegationTokens as well.
3. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12508047/HDFS2665_v1.txt against trunk revision . +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 6 new or modified tests. -1 patch. The patch command could not apply the patch. Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/1732//console> This message is automatically generated.
4. **body:** Not sure why the patch cannot be applied. test-patch results from applying it locally to trunk. -1 overall. +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 6 new or modified tests. -1 javadoc. The javadoc tool appears to have generated 26 warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +1 eclipse:eclipse. The patch built with eclipse:eclipse. -1 findbugs. The patch appears to introduce 13 new Findbugs (version 1.3.9) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. The Findbugs warnings are pre-existing ones. Checking about the javadoc warnings. The patch does contain changes to common. Should that be a separate jira ? Re-uploading the same patch again.
label: code-design
5. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12508130/HDFS2665_v1.txt against trunk revision . +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 6 new or modified tests. -1 patch. The patch command could not apply the patch. Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/1734//console> This message is automatically generated.
6. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12508130/HDFS2665_v1.txt against trunk revision . +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 6 new or modified tests. -1 patch. The patch command could not apply the patch. Console output: <https://builds.apache.org/job/PreCommit-HDFS-Build/1735//console> This message is automatically generated.
7. Sid, I am moving this jira to common and attach common only patch. Please open a separate jira for HDFS part of the patch.
8. Attaching common only part of Sid's patch.
9. -1 overall. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12508137/HADOOP-7933.txt> against trunk revision . +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 3 new or modified tests. -1 javadoc. The javadoc tool appears to have generated 5 warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +1 eclipse:eclipse. The patch built with eclipse:eclipse. +1 findbugs. The patch does not introduce any new Findbugs (version 1.3.9) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. -1 core tests. The patch failed these unit tests: org.apache.hadoop.fs.TestFilterFileSystem +1 contrib tests. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HADOOP-Build/486//testReport/> Console output: <https://builds.apache.org/job/PreCommit-HADOOP-Build/486//console> This message is automatically generated.
10. Updated patch to fix the failing test. Javadoc warnings are not related.

11. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12508168/HADOOP7933_v1.txt against trunk revision . +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 3 new or modified tests. -1 javadoc. The javadoc tool appears to have generated 5 warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +1 eclipse:eclipse. The patch built with eclipse:eclipse. +1 findbugs. The patch does not introduce any new Findbugs (version 1.3.9) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. +1 core tests. The patch passed unit tests in . +1 contrib tests. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HADOOP-Build/487//testReport/> Console output: <https://builds.apache.org/job/PreCommit-HADOOP-Build/487//console> This message is automatically generated.
12. Comments: # FileSystem.java #* Instead of calling credentials.getToken() twice, save it the first time you look and use it later. #* "renewer renewer" drop one of the renewer # ViewFileSystem - unlike FileSystem it does not check if the token already exists in Credentials? If this is the expected behavior, then the javadoc of FileSystem#getDelegationTokens() does not apply to ViewFileSystem?
13. Thanks for the review Suresh. Patch incorporates the feedback. bq. ViewFileSystem - unlike FileSystem it does not check if the token already exists in Credentials? ViewFileSystem does check for existing credentials - so the javadoc continues to apply.
14. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12508321/HADOOP7933_v2.txt against trunk revision . +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 3 new or modified tests. -1 javadoc. The javadoc tool appears to have generated 5 warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +1 eclipse:eclipse. The patch built with eclipse:eclipse. +1 findbugs. The patch does not introduce any new Findbugs (version 1.3.9) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. +1 core tests. The patch passed unit tests in . +1 contrib tests. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HADOOP-Build/489//testReport/> Console output: <https://builds.apache.org/job/PreCommit-HADOOP-Build/489//console> This message is automatically generated.
15. +1 for the change.
16. Thanks for the review Suresh. Committed to trunk and branch-0.23
17. Integrated in Hadoop-Common-trunk-Commit #1487 (See [<https://builds.apache.org/job/Hadoop-Common-trunk-Commit/1487/>]) HADOOP-7933. Add a getDelegationTokens api to FileSystem which checks for known tokens in the passed Credentials object. (sseth) sseth : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1226916> Files : * /hadoop/common/trunk/hadoop-common-project/hadoop-common/CHANGES.txt * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FileSystem.java * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FilterFileSystem.java * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/viewfs/ViewFileSystem.java * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/viewfs/ViewFileSystemBaseTest.java
18. Integrated in Hadoop-Hdfs-trunk-Commit #1559 (See [<https://builds.apache.org/job/Hadoop-Hdfs-trunk-Commit/1559/>]) HADOOP-7933. Add a getDelegationTokens api to FileSystem which checks for known tokens in the passed Credentials object. (sseth) sseth : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1226916> Files : * /hadoop/common/trunk/hadoop-common-project/hadoop-common/CHANGES.txt * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FileSystem.java * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FilterFileSystem.java * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/viewfs/ViewFileSystem.java * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/viewfs/ViewFileSystemBaseTest.java
19. Minor: Maybe store {{mountPoints.get(i).target.targetFileSystem}} in a temp instead of invoking twice? I'm somewhat uneasy about adding a {{FileSystem}} dependency on {{Credentials}}. This forces the caller of {{getDelegationTokens(renewer,creds)}} to use a {{Credentials}} object when ideally the caller should be able to use any object to store the tokens. The method returns a generic {{List}}, so maybe take a generic {{Set}}? (It appears {{Credentials}} was selected for convenience since that's what the MR's

- {{TokenCache}} uses internally?) Would it be reasonable to populate the {{Credentials/Set}} directly? Ie. {{ViewFileSystem}} takes a set of existing tokens, then returns the subset of tokens for its mounts that are not already in the existing tokens. Then the new tokens are added right back into the existing tokens, loop again. I'm concerned that this appears to be creating a cross-component dependency whereby {{ViewFileSystem}} it required to "assume" how MR's {{TokenCache}} will want the tokens keyed within its {{Credentials}}.
20. Darn, missed the commit by minutes. I also noticed {{getDelegationTokens(renewer,creds)}} is returning null where {{getDelegationTokens(renewer)}} would return a 0-length list.
 21. Integrated in Hadoop-Hdfs-0.23-Commit #321 (See [https://builds.apache.org/job/Hadoop-Hdfs-0.23-Commit/321/]) merge HADOOP-7933 from trunk sseth : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1226917> Files : * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/CHANGES.txt * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FileSystem.java * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FilterFileSystem.java * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/viewfs/ViewFileSystem.java * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/viewfs/ViewFileSystemBaseTest.java
 22. Integrated in Hadoop-Common-0.23-Commit #332 (See [https://builds.apache.org/job/Hadoop-Common-0.23-Commit/332/]) merge HADOOP-7933 from trunk sseth : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1226917> Files : * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/CHANGES.txt * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FileSystem.java * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FilterFileSystem.java * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/viewfs/ViewFileSystem.java * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/viewfs/ViewFileSystemBaseTest.java
 23. **body:** {{FileSystem#getDelegationTokens(renewer,creds)}} will leak tokens that already exist, leaving the namenode holding an unnecessary token. Perhaps these should be cancelled.
label: code-design
 24. Integrated in Hadoop-Mapreduce-trunk-Commit #1508 (See [https://builds.apache.org/job/Hadoop-Mapreduce-trunk-Commit/1508/]) HADOOP-7933. Add a getDelegationTokens api to FileSystem which checks for known tokens in the passed Credentials object. (sseth) sseth : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1226916> Files : * /hadoop/common/trunk/hadoop-common-project/hadoop-common/CHANGES.txt * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FileSystem.java * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FilterFileSystem.java * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/viewfs/ViewFileSystem.java * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/viewfs/ViewFileSystemBaseTest.java
 25. Integrated in Hadoop-Mapreduce-0.23-Commit #343 (See [https://builds.apache.org/job/Hadoop-Mapreduce-0.23-Commit/343/]) merge HADOOP-7933 from trunk sseth : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1226917> Files : * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/CHANGES.txt * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FileSystem.java * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FilterFileSystem.java * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/viewfs/ViewFileSystem.java * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/viewfs/ViewFileSystemBaseTest.java
 26. bq. Minor: Maybe store mountPoints.get.target.targetFileSystem in a temp instead of invoking twice? Could be done - but i don't believe this lookup is very expensive. bq. It appears Credentials was selected for convenience since that's what the MR's TokenCache uses internally? It appears Credentials was

selected for convenience since that's what the MR's TokenCache uses internally? The API is limited private (at least for now) to MR and HDFS - so I don't see this as a very big issue. Changing it to a set is a simple change though. Thoughts ? bq. Would it be reasonable to populate the Credentials/Set directly? Ie. ViewFileSystem takes a set of existing tokens, then returns the subset of tokens for its mounts that are not already in the existing tokens. Was returning all tokens (instead of just new ones) to be consistent with the existing API. Having this call populate tokens into the Credentials object was an option - but that increases the cross-component dependency. bq. I'm concerned that this appears to be creating a cross-component dependency whereby ViewFileSystem it required to "assume" how MR's TokenCache will want the tokens keyed within its Credentials. I'm not very happy with this either - suggestion on a better approach ? Again, being LimitedPrivate to MR and HDFS limits the impact. bq. I also noticed getDelegationTokens(renewer,creds) is returning null where getDelegationTokens(renewer) would return a 0-length list. I believe getDelegationTokens(renewer, creds) is also returning an empty list and not null. However, there is a potential issue with the ViewFileSystem implementation - {{result = new ArrayList<Token<?>>(initialListSize)}} - if there's multiple mount points - part of the array will not be populated.

27. Integrated in Hadoop-Hdfs-trunk #915 (See [<https://builds.apache.org/job/Hadoop-Hdfs-trunk/915/>]) HADOOP-7933. Add a getDelegationTokens api to FileSystem which checks for known tokens in the passed Credentials object. (sseth) sseth : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1226916> Files : * /hadoop/common/trunk/hadoop-common-project/hadoop-common/CHANGES.txt * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FileSystem.java * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FilterFileSystem.java * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/viewfs/ViewFileSystem.java * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/viewfs/ViewFileSystemBaseTest.java
28. Integrated in Hadoop-Hdfs-0.23-Build #128 (See [<https://builds.apache.org/job/Hadoop-Hdfs-0.23-Build/128/>]) merge HADOOP-7933 from trunk sseth : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1226917> Files : * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/CHANGES.txt * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FileSystem.java * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FilterFileSystem.java * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/viewfs/ViewFileSystem.java * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/viewfs/ViewFileSystemBaseTest.java
29. Integrated in Hadoop-Mapreduce-0.23-Build #150 (See [<https://builds.apache.org/job/Hadoop-Mapreduce-0.23-Build/150/>]) merge HADOOP-7933 from trunk sseth : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1226917> Files : * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/CHANGES.txt * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FileSystem.java * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FilterFileSystem.java * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/viewfs/ViewFileSystem.java * /hadoop/common/branches/branch-0.23/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/viewfs/ViewFileSystemBaseTest.java
30. Integrated in Hadoop-Mapreduce-trunk #948 (See [<https://builds.apache.org/job/Hadoop-Mapreduce-trunk/948/>]) HADOOP-7933. Add a getDelegationTokens api to FileSystem which checks for known tokens in the passed Credentials object. (sseth) sseth : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1226916> Files : * /hadoop/common/trunk/hadoop-common-project/hadoop-common/CHANGES.txt * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FileSystem.java * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FilterFileSystem.java * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/viewfs/ViewFileSystem.java * /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/viewfs/ViewFileSystemBaseTest.java

31. {quote} bq. It appears Credentials was selected for convenience since that's what the MR's TokenCache uses internally? It appears Credentials was selected for convenience since that's what the MR's TokenCache uses internally? The API is limited private (at least for now) to MR and HDFS - so I don't see this as a very big issue. Changing it to a set is a simple change though. Thoughts ? {quote} Although it's limited audience, it's arguably creating a cross-component dependency where the cart is leading the horse. However, {{Credentials}} is part of common, so perhaps it's not a big deal or even moot if the next point is addressed. {quote} bq. I'm concerned that this appears to be creating a cross-component dependency whereby ViewFileSystem it required to "assume" how MR's TokenCache will want the tokens keyed within its Credentials. I'm not very happy with this either - suggestion on a better approach ? Again, being LimitedPrivate to MR and HDFS limits the impact. {quote} A simple solution is to centralize the chosen key to a single location. Awhile back I played with adding {{Credentials#addToken(Token)}} or {{Credentials#addServiceToken(token)}} to enforce {{token.getService()}} as the key. Along the same lines, {{Credentials#getToken(TokenIssuer)}} could fetch all tokens for the token issuing object. The cited inconsistency about {{getDelegationTokens}} was a misread of the code...
32. Integrated in Hadoop-Hdfs-trunk-Commit #1573 (See [<https://builds.apache.org/job/Hadoop-Hdfs-trunk-Commit/1573/>]) HDFS-2710. Add HDFS tests related to HADOOP-7933. Contributed by Siddarth Seth. suresh : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1227756> Files : * /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt * /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/viewfs/TestViewFileSystemHdfs.java
33. Integrated in Hadoop-Common-trunk-Commit #1501 (See [<https://builds.apache.org/job/Hadoop-Common-trunk-Commit/1501/>]) HDFS-2710. Add HDFS tests related to HADOOP-7933. Contributed by Siddarth Seth. suresh : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1227756> Files : * /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt * /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/viewfs/TestViewFileSystemHdfs.java
34. Integrated in Hadoop-Mapreduce-trunk-Commit #1521 (See [<https://builds.apache.org/job/Hadoop-Mapreduce-trunk-Commit/1521/>]) HDFS-2710. Add HDFS tests related to HADOOP-7933. Contributed by Siddarth Seth. suresh : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1227756> Files : * /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt * /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/viewfs/TestViewFileSystemHdfs.java
35. Integrated in Hadoop-Hdfs-trunk #917 (See [<https://builds.apache.org/job/Hadoop-Hdfs-trunk/917/>]) HDFS-2710. Add HDFS tests related to HADOOP-7933. Contributed by Siddarth Seth. suresh : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1227756> Files : * /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt * /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/viewfs/TestViewFileSystemHdfs.java
36. Integrated in Hadoop-Mapreduce-trunk #950 (See [<https://builds.apache.org/job/Hadoop-Mapreduce-trunk/950/>]) HDFS-2710. Add HDFS tests related to HADOOP-7933. Contributed by Siddarth Seth. suresh : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1227756> Files : * /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/CHANGES.txt * /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/viewfs/TestViewFileSystemHdfs.java