

Item 155

git_comments:

1. if all maps are assigned, then ramp up all reduces irrespective of the headroom
2. verify slow-start still in effect when no more maps need to be scheduled but some have yet to complete

git_commits:

1. **summary:** svn merge -c 1354181 FIXES: MAPREDUCE-4228.
mapreduce.job.reduce.slowstart.completedmaps is not working properly (Jason Lowe via bobby)
message: svn merge -c 1354181 FIXES: MAPREDUCE-4228.
mapreduce.job.reduce.slowstart.completedmaps is not working properly (Jason Lowe via bobby) git-svn-id: <https://svn.apache.org/repos/asf/hadoop/common/branches/branch-0.23@1354185> 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** mapreduce.job.reduce.slowstart.completedmaps is not working properly to delay the scheduling of the reduce tasks
description: If no more map tasks need to be scheduled but not all have completed, the ApplicationMaster will start scheduling reducers even if the number of completed maps has not met the mapreduce.job.reduce.slowstart.completedmaps threshold. For example, if the property is set to 1.0 all maps should complete before any reducers are scheduled. However the reducers are scheduled as soon as the last map task is assigned to a container. For a job with very long-running maps, a cluster with enough capacity to launch all map tasks could cause reducers to launch prematurely and waste cluster resources. Thanks to Phil Su for discovering this issue.
label: code-design
2. **summary:** mapreduce.job.reduce.slowstart.completedmaps is not working properly to delay the scheduling of the reduce tasks
description: If no more map tasks need to be scheduled but not all have completed, the ApplicationMaster will start scheduling reducers even if the number of completed maps has not met the mapreduce.job.reduce.slowstart.completedmaps threshold. For example, if the property is set to 1.0 all maps should complete before any reducers are scheduled. However the reducers are scheduled as soon as the last map task is assigned to a container. For a job with very long-running maps, a cluster with enough capacity to launch all map tasks could cause reducers to launch prematurely and waste cluster resources. Thanks to Phil Su for discovering this issue.
3. **summary:** mapreduce.job.reduce.slowstart.completedmaps is not working properly to delay the scheduling of the reduce tasks
description: If no more map tasks need to be scheduled but not all have completed, the ApplicationMaster will start scheduling reducers even if the number of completed maps has not met the mapreduce.job.reduce.slowstart.completedmaps threshold. For example, if the property is set to 1.0 all maps should complete before any reducers are scheduled. However the reducers are scheduled as soon as the last map task is assigned to a container. For a job with very long-running maps, a cluster with enough capacity to launch all map tasks could cause reducers to launch prematurely and waste cluster resources. Thanks to Phil Su for discovering this issue.
label: code-design
4. **summary:** mapreduce.job.reduce.slowstart.completedmaps is not working properly to delay the scheduling of the reduce tasks

description: If no more map tasks need to be scheduled but not all have completed, the ApplicationMaster will start scheduling reducers even if the number of completed maps has not met the `mapreduce.job.reduce.slowstart.completedmaps` threshold. For example, if the property is set to 1.0 all maps should complete before any reducers are scheduled. However the reducers are scheduled as soon as the last map task is assigned to a container. For a job with very long-running maps, a cluster with enough capacity to launch all map tasks could cause reducers to launch prematurely and waste cluster resources. Thanks to Phil Su for discovering this issue.

5. **summary:** `mapreduce.job.reduce.slowstart.completedmaps` is not working properly to delay the scheduling of the reduce tasks

description: If no more map tasks need to be scheduled but not all have completed, the ApplicationMaster will start scheduling reducers even if the number of completed maps has not met the `mapreduce.job.reduce.slowstart.completedmaps` threshold. For example, if the property is set to 1.0 all maps should complete before any reducers are scheduled. However the reducers are scheduled as soon as the last map task is assigned to a container. For a job with very long-running maps, a cluster with enough capacity to launch all map tasks could cause reducers to launch prematurely and waste cluster resources. Thanks to Phil Su for discovering this issue.

6. **summary:** `mapreduce.job.reduce.slowstart.completedmaps` is not working properly to delay the scheduling of the reduce tasks

description: If no more map tasks need to be scheduled but not all have completed, the ApplicationMaster will start scheduling reducers even if the number of completed maps has not met the `mapreduce.job.reduce.slowstart.completedmaps` threshold. For example, if the property is set to 1.0 all maps should complete before any reducers are scheduled. However the reducers are scheduled as soon as the last map task is assigned to a container. For a job with very long-running maps, a cluster with enough capacity to launch all map tasks could cause reducers to launch prematurely and waste cluster resources. Thanks to Phil Su for discovering this issue.

7. **summary:** `mapreduce.job.reduce.slowstart.completedmaps` is not working properly to delay the scheduling of the reduce tasks

description: If no more map tasks need to be scheduled but not all have completed, the ApplicationMaster will start scheduling reducers even if the number of completed maps has not met the `mapreduce.job.reduce.slowstart.completedmaps` threshold. For example, if the property is set to 1.0 all maps should complete before any reducers are scheduled. However the reducers are scheduled as soon as the last map task is assigned to a container. For a job with very long-running maps, a cluster with enough capacity to launch all map tasks could cause reducers to launch prematurely and waste cluster resources. Thanks to Phil Su for discovering this issue.

8. **summary:** `mapreduce.job.reduce.slowstart.completedmaps` is not working properly to delay the scheduling of the reduce tasks

description: If no more map tasks need to be scheduled but not all have completed, the ApplicationMaster will start scheduling reducers even if the number of completed maps has not met the `mapreduce.job.reduce.slowstart.completedmaps` threshold. For example, if the property is set to 1.0 all maps should complete before any reducers are scheduled. However the reducers are scheduled as soon as the last map task is assigned to a container. For a job with very long-running maps, a cluster with enough capacity to launch all map tasks could cause reducers to launch prematurely and waste cluster resources. Thanks to Phil Su for discovering this issue.

9. **summary:** `mapreduce.job.reduce.slowstart.completedmaps` is not working properly to delay the scheduling of the reduce tasks

description: If no more map tasks need to be scheduled but not all have completed, the ApplicationMaster will start scheduling reducers even if the number of completed maps has not met the `mapreduce.job.reduce.slowstart.completedmaps` threshold. For example, if the property is set to 1.0 all maps should complete before any reducers are scheduled. However the reducers are scheduled as soon as the last map task is assigned to a container. For a job with very long-running maps, a cluster with enough capacity to launch all map tasks could cause reducers to launch prematurely and waste cluster resources. Thanks to Phil Su for discovering this issue.

label: code-design

10. **summary:** `mapreduce.job.reduce.slowstart.completedmaps` is not working properly to delay the scheduling of the reduce tasks

description: If no more map tasks need to be scheduled but not all have completed, the ApplicationMaster will start scheduling reducers even if the number of completed maps has not met the `mapreduce.job.reduce.slowstart.completedmaps` threshold. For example, if the property is set to 1.0 all maps should complete before any reducers are scheduled. However the reducers are scheduled as soon as

the last map task is assigned to a container. For a job with very long-running maps, a cluster with enough capacity to launch all map tasks could cause reducers to launch prematurely and waste cluster resources. Thanks to Phil Su for discovering this issue.

11. **summary:** `mapreduce.job.reduce.slowstart.completedmaps` is not working properly to delay the scheduling of the reduce tasks

description: If no more map tasks need to be scheduled but not all have completed, the ApplicationMaster will start scheduling reducers even if the number of completed maps has not met the `mapreduce.job.reduce.slowstart.completedmaps` threshold. For example, if the property is set to 1.0 all maps should complete before any reducers are scheduled. However the reducers are scheduled as soon as the last map task is assigned to a container. For a job with very long-running maps, a cluster with enough capacity to launch all map tasks could cause reducers to launch prematurely and waste cluster resources. Thanks to Phil Su for discovering this issue.

12. **summary:** `mapreduce.job.reduce.slowstart.completedmaps` is not working properly to delay the scheduling of the reduce tasks

description: If no more map tasks need to be scheduled but not all have completed, the ApplicationMaster will start scheduling reducers even if the number of completed maps has not met the `mapreduce.job.reduce.slowstart.completedmaps` threshold. For example, if the property is set to 1.0 all maps should complete before any reducers are scheduled. However the reducers are scheduled as soon as the last map task is assigned to a container. For a job with very long-running maps, a cluster with enough capacity to launch all map tasks could cause reducers to launch prematurely and waste cluster resources. Thanks to Phil Su for discovering this issue.

13. **summary:** `mapreduce.job.reduce.slowstart.completedmaps` is not working properly to delay the scheduling of the reduce tasks

description: If no more map tasks need to be scheduled but not all have completed, the ApplicationMaster will start scheduling reducers even if the number of completed maps has not met the `mapreduce.job.reduce.slowstart.completedmaps` threshold. For example, if the property is set to 1.0 all maps should complete before any reducers are scheduled. However the reducers are scheduled as soon as the last map task is assigned to a container. For a job with very long-running maps, a cluster with enough capacity to launch all map tasks could cause reducers to launch prematurely and waste cluster resources. Thanks to Phil Su for discovering this issue.

14. **summary:** `mapreduce.job.reduce.slowstart.completedmaps` is not working properly to delay the scheduling of the reduce tasks

description: If no more map tasks need to be scheduled but not all have completed, the ApplicationMaster will start scheduling reducers even if the number of completed maps has not met the `mapreduce.job.reduce.slowstart.completedmaps` threshold. For example, if the property is set to 1.0 all maps should complete before any reducers are scheduled. However the reducers are scheduled as soon as the last map task is assigned to a container. For a job with very long-running maps, a cluster with enough capacity to launch all map tasks could cause reducers to launch prematurely and waste cluster resources. Thanks to Phil Su for discovering this issue.

15. **summary:** `mapreduce.job.reduce.slowstart.completedmaps` is not working properly to delay the scheduling of the reduce tasks

description: If no more map tasks need to be scheduled but not all have completed, the ApplicationMaster will start scheduling reducers even if the number of completed maps has not met the `mapreduce.job.reduce.slowstart.completedmaps` threshold. For example, if the property is set to 1.0 all maps should complete before any reducers are scheduled. However the reducers are scheduled as soon as the last map task is assigned to a container. For a job with very long-running maps, a cluster with enough capacity to launch all map tasks could cause reducers to launch prematurely and waste cluster resources. Thanks to Phil Su for discovering this issue.

16. **summary:** `mapreduce.job.reduce.slowstart.completedmaps` is not working properly to delay the scheduling of the reduce tasks

description: If no more map tasks need to be scheduled but not all have completed, the ApplicationMaster will start scheduling reducers even if the number of completed maps has not met the `mapreduce.job.reduce.slowstart.completedmaps` threshold. For example, if the property is set to 1.0 all maps should complete before any reducers are scheduled. However the reducers are scheduled as soon as the last map task is assigned to a container. For a job with very long-running maps, a cluster with enough capacity to launch all map tasks could cause reducers to launch prematurely and waste cluster resources. Thanks to Phil Su for discovering this issue.

1. **body:** It's a little confusing, if `mapreduce.job.reduce.slowstart.completedmaps` didn't control when reduce will start scheduling, should we just remove this parameter? or maybe change name to `mapreduce.job.reduce.slowstart.startedmaps`?
label: code-design
2. As I understand it, this functionality works properly in Hadoop 1.0, so that reducers do not start until the requisite number of map tasks have completed. In 0.23/2.0 that's not the case, and we need to correct that behavior rather than rename existing properties to match an undesired behavior.
3. **body:** this is mainly for small jobs, isn't it? the time frame between started map and completed map job will affect the small job's latency, if that's the case, why not just remove this parameter, what's the difference if we set it to 0.33 and 1.0? we didn't prevent other reduce to locate, isn't it?
label: code-design
4. This is mostly about the utilization of the cluster. In many cases you have a map/reduce job where the mappers take a very long time. If you don't wait until all of the mappers have finished there are many many reducers taking up resources doing nothing waiting for the mappers to finish. In some cases we have seen this last from several hours up to a day or two.
5. Patch to verify we've met the slowstart criteria before unleashing all reducers once all maps have been assigned. Besides the automated tests that were added/modified, I also ran manual tests on a local cluster with sleep and wordcount jobs.
6. -1 overall. Here are the results of testing the latest attachment
<http://issues.apache.org/jira/secure/attachment/12525892/MAPREDUCE-4228.patch> against trunk revision . +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 1 new or modified test files. -1 javadoc. The javadoc tool appears to have generated 2 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 hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-app. +1 contrib tests. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/2362//testReport/> Console output: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/2362//console> This message is automatically generated.
7. Minor patch update to add Private annotation on `scheduleAllReduces()` which was made public for testing.
8. -1 overall. Here are the results of testing the latest attachment
<http://issues.apache.org/jira/secure/attachment/12525912/MAPREDUCE-4228.patch> against trunk revision . +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 1 new or modified test files. -1 javadoc. The javadoc tool appears to have generated 2 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 cause Findbugs (version 1.3.9) to fail. +1 release audit. The applied patch does not increase the total number of release audit warnings. +1 core tests. The patch passed unit tests in hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-app. +1 contrib tests. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/2365//testReport/> Console output: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/2365//console> This message is automatically generated.
9. **body:** The javadoc warnings are unrelated. Not sure why findbugs failed, as it ran just fine for me locally (and on the first patch version). Output from my local run: -1 overall. +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 1 new or modified test files. -1 javadoc. The javadoc tool appears to have generated 2 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) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings.
label: code-design
10. Upmerged patch to latest changes on trunk.
11. -1 overall. Here are the results of testing the latest attachment
<http://issues.apache.org/jira/secure/attachment/12532955/MAPREDUCE-4228.patch> against trunk revision . +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 1 new or modified test files. +1 javac. The applied patch does not increase the total number of javac compiler warnings. -1 javadoc. The javadoc tool appears to have generated 13 warning messages. +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 hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-app. +1 contrib tests. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/2494/testReport/> Console output: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/2494/console> This message is automatically generated.
12. Javadoc warnings are a known issue. See HDFS-3550.
 13. Thanks Jason the changes look good to me +1.
 14. Thanks for the patch Jason, I have put this into trunk, branch-2, and branch-0.23.
 15. Integrated in Hadoop-Common-trunk-Commit #2391 (See [<https://builds.apache.org/job/Hadoop-Common-trunk-Commit/2391/>]) MAPREDUCE-4228. mapreduce.job.reduce.slowstart.completedmaps is not working properly (Jason Lowe via bobby) (Revision 1354181) Result = SUCCESS bobby : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1354181> Files : *
/hadoop/common/trunk/hadoop-mapreduce-project/CHANGES.txt * /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-app/src/main/java/org/apache/hadoop/mapreduce/v2/app/rm/RMContainerAllocator.java *
/hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-app/src/test/java/org/apache/hadoop/mapreduce/v2/app/TestRMContainerAllocator.java
 16. Integrated in Hadoop-Hdfs-trunk-Commit #2460 (See [<https://builds.apache.org/job/Hadoop-Hdfs-trunk-Commit/2460/>]) MAPREDUCE-4228. mapreduce.job.reduce.slowstart.completedmaps is not working properly (Jason Lowe via bobby) (Revision 1354181) Result = SUCCESS bobby : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1354181> Files : *
/hadoop/common/trunk/hadoop-mapreduce-project/CHANGES.txt * /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-app/src/main/java/org/apache/hadoop/mapreduce/v2/app/rm/RMContainerAllocator.java *
/hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-app/src/test/java/org/apache/hadoop/mapreduce/v2/app/TestRMContainerAllocator.java
 17. Integrated in Hadoop-Mapreduce-trunk-Commit #2410 (See [<https://builds.apache.org/job/Hadoop-Mapreduce-trunk-Commit/2410/>]) MAPREDUCE-4228. mapreduce.job.reduce.slowstart.completedmaps is not working properly (Jason Lowe via bobby) (Revision 1354181) Result = FAILURE bobby : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1354181> Files : *
/hadoop/common/trunk/hadoop-mapreduce-project/CHANGES.txt * /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-app/src/main/java/org/apache/hadoop/mapreduce/v2/app/rm/RMContainerAllocator.java *
/hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-app/src/test/java/org/apache/hadoop/mapreduce/v2/app/TestRMContainerAllocator.java
 18. Integrated in Hadoop-Hdfs-trunk #1089 (See [<https://builds.apache.org/job/Hadoop-Hdfs-trunk/1089/>]) MAPREDUCE-4228. mapreduce.job.reduce.slowstart.completedmaps is not working properly (Jason Lowe via bobby) (Revision 1354181) Result = FAILURE bobby : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1354181> Files : * /hadoop/common/trunk/hadoop-mapreduce-project/CHANGES.txt * /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-app/src/main/java/org/apache/hadoop/mapreduce/v2/app/rm/RMContainerAllocator.java *
/hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-app/src/test/java/org/apache/hadoop/mapreduce/v2/app/TestRMContainerAllocator.java
 19. Integrated in Hadoop-Hdfs-0.23-Build #299 (See [<https://builds.apache.org/job/Hadoop-Hdfs-0.23-Build/299/>]) svn merge -c 1354181 FIXES: MAPREDUCE-4228.
mapreduce.job.reduce.slowstart.completedmaps is not working properly (Jason Lowe via bobby) (Revision 1354185) Result = UNSTABLE bobby : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1354185> Files : * /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/CHANGES.txt * /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-app/src/main/java/org/apache/hadoop/mapreduce/v2/app/rm/RMContainerAllocator.java *
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-app/src/test/java/org/apache/hadoop/mapreduce/v2/app/TestRMContainerAllocator.java
 20. Integrated in Hadoop-Mapreduce-trunk #1122 (See [<https://builds.apache.org/job/Hadoop-Mapreduce-trunk/1122/>]) MAPREDUCE-4228. mapreduce.job.reduce.slowstart.completedmaps is not working properly (Jason Lowe via bobby) (Revision 1354181) Result = FAILURE bobby : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1354181> Files : *

```
/hadoop/common/trunk/hadoop-mapreduce-project/CHANGES.txt * /hadoop/common/trunk/hadoop-  
mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-  
app/src/main/java/org/apache/hadoop/mapreduce/v2/app/rm/RMContainerAllocator.java *  
/hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-  
app/src/test/java/org/apache/hadoop/mapreduce/v2/app/TestRMContainerAllocator.java
```