

git_comments:

1. Bump up the pool size to `idealPoolSize+INITIAL_POOL_SIZE`, the later is just a buffer so we are not always increasing the pool-size
2. Start with a default core-pool size of 10 and change it dynamically.
3. See if we need up the pool size only if haven't reached the maximum limit yet.
4. nodes where containers will run at **this** point of time. This is **not** the cluster size and doesn't need to be.
5. **** Upper limit on the number of threads user to launch containers in the app *** master. Expect level config, you shouldn't be needing it in most cases.

git_commits:

1. **summary:** Merge -r 1175293:1175294 from trunk to branch-0.23 to fix MAPREDUCE-2961.
message: Merge -r 1175293:1175294 from trunk to branch-0.23 to fix MAPREDUCE-2961. git-svn-id: <https://svn.apache.org/repos/asf/hadoop/common/branches/branch-0.23@1175295> 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Increase the default threadpool size for container launching in the application master.
description: Currently the default threadpool size is 10 for launching containers in `ContainerLauncherImpl`. We should increase that to 100 for a reasonable default, so that container launching is not backed up by a small thread pool size.

jira_issues_comments:

1. Trivial patch to increase the number of threads in the launcher so that we do not bottleneck on container launching.
2. -1 overall. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12493691/MAPREDUCE-2961.patch> against trunk revision . +1 @author. The patch does not contain any @author tags. -1 tests included. The patch doesn't appear to include any new or modified tests. Please justify why no new tests are needed for this patch. Also please list what manual steps were performed to verify this patch. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +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-MAPREDUCE-Build/665/testReport/> Findbugs warnings: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/665/artifact/trunk/patchprocess/newPatchFindbugsWarningshadoop-mapreduce-client-hs.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/665/artifact/trunk/patchprocess/newPatchFindbugsWarningshadoop-mapreduce-client-shuffle.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/665/artifact/trunk/patchprocess/newPatchFindbugsWarningshadoop-mapreduce-client-common.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/665/artifact/trunk/patchprocess/newPatchFindbugsWarningshadoop-mapreduce-client-app.html> Findbugs warnings: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/665/artifact/trunk/patchprocess/newPatchFindbugsWarningshadoop-mapreduce-client->

jobclient.html Findbugs warnings: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/665//artifact/trunk/patchprocess/newPatchFindbugsWarningshadoop-mapreduce-client-core.html>
Findbugs warnings: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/665//artifact/trunk/patchprocess/newPatchFindbugsWarningshadoop-yarn-server-common.html>
Findbugs warnings: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/665//artifact/trunk/patchprocess/newPatchFindbugsWarningshadoop-yarn-server-resourcemanager.html>
Findbugs warnings: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/665//artifact/trunk/patchprocess/newPatchFindbugsWarningshadoop-yarn-server-nodemanager.html>
Findbugs warnings: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/665//artifact/trunk/patchprocess/newPatchFindbugsWarningshadoop-yarn-common.html>
Findbugs warnings: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/665//artifact/trunk/patchprocess/newPatchFindbugsWarningshadoop-yarn-api.html>
Console output: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/665//console> This message is automatically generated.

3. Mahadev, you need to rebase after MAPREDUCE-2864. Thanks.
4. I definitely think that instead of a config, this should be a function of #nodes and #tasks with defaults to 10. Will provide a patch for the same.
5. Vinod, what heuristic do you have in mind?
6. the same patch on top of new changes.
7. @Vinod, I do agree this is a hack but unless we see that it doesnt work, I probably think we should not spend too much time on estimating the right amount of threads. Thoughts?
8. -1 overall. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12496181/MAPREDUCE-2961.patch> against trunk revision . +1 @author. The patch does not contain any @author tags. -1 tests included. The patch doesn't appear to include any new or modified tests. Please justify why no new tests are needed for this patch. Also please list what manual steps were performed to verify this patch. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +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-MAPREDUCE-Build/831//testReport/>
Console output: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/831//console> This message is automatically generated.
9. Forgot about this one. It is easy to set the number of threads dynamically. We just need to set it equal to the number of nodes in the system. We are looking at a scalability of 10,000 nodes, so I'd put a cap of 2000 threads, hardcoded.
10. There are a couple of other bugs that need to be fixed related to ContainerLauncher's thread pool. - The way it is, the pool size grows to the configured core-pool-size and stays there. We should set core-pool-size to be very low, set a max-pool-size and bound the queue size. - We also pre-allocate threads which affects MRAppMaster initialization time. I am taking this up.
11. Attaching patch that should fix this. bq. The way it is, the pool size grows to the configured core-pool-size and stays there. We should set core-pool-size to be very low, set a max-pool-size and bound the queue size. We don't need to do this anymore, as the corepool itself is changed according to container-demand.
12. nice patch, looks good to me! Will commit after hudson +1's.
13. Vinod, Doesnt look like there is cap on the thread pool size. Should we put in cap at 300-400 or so?
14. Anything above 100 threads in a overkill I think... plus each thread has a stack of 1MB. So, we shouldn't be using too many.
15. -1 overall. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12496232/MAPREDUCE-2961-20110923.txt> against trunk revision . +1 @author. The patch does not contain any @author tags. -1 tests included. The patch doesn't appear to include any new or modified tests. Please justify why no new tests are needed for this patch. Also please list what manual steps were performed to verify this patch. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +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-MAPREDUCE-Build/835//testReport/>

- Console output: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/835//console> This message is automatically generated.
16. Updating patch, including the limit on the number of threads. With this patch, in most cases, the thread-pool should scale according to the need.
 17. For very large clusters, one can up the limit(default 500) to increase scalability via an expert-level, non-documented configuration. The default stack-size for a thread is 500K, so given 2GB default vmem for AM, default upper limit of 500 should be fine.
 18. Oh, BTW, I took Karam's help to see this patch's effect. Before this patch, we had 200 core-pool size for a 100,000 map-only jobs on ~350 nodes. With this patch, which automatically takes care of the thread-pool size(and sets to ~350 for this job in particular), we saw ~30% improvement on wall-clock time.
 19. -1 overall. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12496276/MAPREDUCE-2961-20110923.1.txt> against trunk revision . +1 @author. The patch does not contain any @author tags. -1 tests included. The patch doesn't appear to include any new or modified tests. Please justify why no new tests are needed for this patch. Also please list what manual steps were performed to verify this patch. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +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-MAPREDUCE-Build/839//testReport/> Console output: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/839//console> This message is automatically generated.
 20. Minor change - move default value to MRJobConfig.
 21. I just committed this. Thanks Vinod!
 22. -1 overall. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12496379/MAPREDUCE-2961.patch> against trunk revision . +1 @author. The patch does not contain any @author tags. -1 tests included. The patch doesn't appear to include any new or modified tests. Please justify why no new tests are needed for this patch. Also please list what manual steps were performed to verify this patch. -1 patch. The patch command could not apply the patch. Console output: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/851//console> This message is automatically generated.
 23. Integrated in Hadoop-Hdfs-0.23-Build #20 (See [<https://builds.apache.org/job/Hadoop-Hdfs-0.23-Build/20/>]) Merge -r 1175293:1175294 from trunk to branch-0.23 to fix MAPREDUCE-2961. acmurthy : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1175295> 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/launcher/ContainerLauncherImpl.java *
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/MRJobConfig.java
 24. Integrated in Hadoop-Mapreduce-0.23-Build #25 (See [<https://builds.apache.org/job/Hadoop-Mapreduce-0.23-Build/25/>]) Merge -r 1175293:1175294 from trunk to branch-0.23 to fix MAPREDUCE-2961. acmurthy : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1175295> 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/launcher/ContainerLauncherImpl.java *
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/MRJobConfig.java