Hadoop Map/Reduce / MAPREDUCE-6514

Job hangs as ask is not updated after ramping down of all reducers

Details

Type:	Bug	Status:	CLOSED
Priority:	Blocker	Resolution:	Fixed
Affects Version/s:	None	Fix Version/s:	2.8.0, 2.7.3, 2.6.5, 3.0.0-alpha1
Component/s:	applicationmaster		
Labels:	None		
Target Version/s:	2.8.0, 2.7.3, 2.6.4		
Hadoop Flags:	Reviewed		

Description

In RMContainerAllocator#preemptReducesIfNeeded, we simply clear the scheduled reduces map and put these reducers to pending. This is not updated in ask. So RM keeps on assigning and AM is not able to assign as no reducer is scheduled(check logs below the code).
If this is updated immediately, RM will be able to schedule mappers immediately which anyways is the intention when we ramp down reducers.
Scheduler need not allocate for ramped down reducers
This if not handled can lead to map starvation as pointed out in ~~MAPREDUCE-6513~~

```
LOG.info("Ramping down all scheduled reduces:"
           + scheduledRequests.reduces.size());
       for (ContainerRequest req : scheduledRequests.reduces.values()) {
         pendingReduces.add(req);
       }
       scheduledRequests.reduces.clear();
```

```
2015-10-13 04:55:04,912 INFO [RMCommunicator Allocator] org.apache.hadoop.mapreduce.v2.app.rm.RMContainerAllocator: Container n
2015-10-13 04:55:04,912 INFO [RMCommunicator Allocator] org.apache.hadoop.mapreduce.v2.app.rm.RMContainerAllocator: Cannot assi
2015-10-13 04:55:04,912 INFO [RMCommunicator Allocator] org.apache.hadoop.mapreduce.v2.app.rm.RMContainerAllocator: Container n
2015-10-13 04:55:04,912 INFO [RMCommunicator Allocator] org.apache.hadoop.mapreduce.v2.app.rm.RMContainerAllocator: Cannot assi
```

Attachments

MAPREDUCE-6514.01.patch	10 kB	29/Oct/15 20:01
MAPREDUCE-6514.02.patch	11 kB	05/May/16 18:25

Issue Links

is broken by

MAPREDUCE-6302 Preempt reducers after a configurable timeout irrespective of headroom CLOSED

is related to

MAPREDUCE-6513 MR job got hanged forever when one NM unstable for some time CLOSED

Activity

Rohith Sharma K S added a comment - 17/Oct/15 07:11

Thanks varun_saxena for creating issue.

I too have seen many times this particular problem because of not updating ResourceRequest when Ramping down the reducers. As a result, number of containers count in the ask is increased as explained in the below

Consider MR job is running with 3 mapper and 100 reducers. All mappers were finished, and 3 reducers are running in the cluster. Kill the task attempt of mapper. So Ramping down will happen.

```
2015-10-17 12:09:10,836 INFO [RMCommunicator Allocator] org.apache.hadoop.mapreduce.v2.app.rm.RMContainerAllocator: Rampir
```

Since resourcerequest is not decremented from remoteRequestsTable. Next time when ask for 97 reducers i.e **pendingReducers --> scheduledReducers**, which starts adding number of containers to existing RR. So `97(previous outstanding reducers)+100(total reducers)=197`

```
2015-10-17 12:10:49,617 INFO [RMCommunicator Allocator] org.apache.hadoop.mapreduce.v2.app.rm.RMContainerAllocator: All ma
2015-10-17 12:10:49,617 DEBUG [RMCommunicator Allocator] org.apache.hadoop.mapreduce.v2.app.rm.RMContainerRequestor: addRe
2015-10-17 12:10:49,617 DEBUG [RMCommunicator Allocator] org.apache.hadoop.mapreduce.v2.app.rm.RMContainerRequestor: addRe
2015-10-17 12:10:49,617 DEBUG [RMCommunicator Allocator] org.apache.hadoop.mapreduce.v2.app.rm.RMContainerRequestor: addRe
......
......
2015-10-17 12:10:49,647 DEBUG [RMCommunicator Allocator] org.apache.hadoop.mapreduce.v2.app.rm.RMContainerRequestor: addRe
2015-10-17 12:10:49,648 DEBUG [RMCommunicator Allocator] org.apache.hadoop.mapreduce.v2.app.rm.RMContainerRequestor: addRe
2015-10-17 12:10:49,648 DEBUG [RMCommunicator Allocator] org.apache.hadoop.mapreduce.v2.app.rm.RMContainerRequestor: addRe
```

In ResourceManager.log
In the log, observe that **Containers: 100** before ramping down all the reducers. Afterwords, MR ask for reducers requests, **Containers: 197**

```
2015-10-17 12:07:40,613 DEBUG org.apache.hadoop.yarn.server.resourcemanager.scheduler.SchedulerApplicationAttempt: showReq
```

After ramping down all the reducers and request for ask.

```
2015-10-17 12:10:50,651 DEBUG org.apache.hadoop.yarn.server.resourcemanager.scheduler.SchedulerApplicationAttempt: showReq
```

Basically it causes unnecessary overhead on ResourceManager that for assignment and also effect other applications container assignment. These allocated containers are rejected by MR.

Sunil G added a comment - 17/Oct/15 09:15

Yes rohithsharma, thanks for the detailed analysis. This improvement can definitely avoid many unnecessary allocations from RM side.
And if any extra container allocation happened from RM side within a heartbeat interval (AM requested less number of reducers, however more containers were already allocated by RM), such containers has to be rejected by AM and we already do that. I hope this number wont be big as it may happen only within a heartbeat interval.

Hadoop QA added a comment - 29/Oct/15 21:48

☒ **-1 overall**

Vote	Subsystem	Runtime	Comment
0	reexec	0m 6s	docker + precommit patch detected.
+1	@author	0m 0s	The patch does not contain any @author tags.
+1	test4tests	0m 0s	The patch appears to include 1 new or modified test files.
+1	mvninstall	3m 30s	trunk passed
+1	compile	0m 17s	trunk passed with JDK v1.8.0_60
+1	compile	0m 19s	trunk passed with JDK v1.7.0_79
+1	checkstyle	0m 12s	trunk passed
+1	mvneclipse	0m 13s	trunk passed
+1	findbugs	0m 45s	trunk passed
+1	javadoc	0m 16s	trunk passed with JDK v1.8.0_60
+1	javadoc	0m 17s	trunk passed with JDK v1.7.0_79
+1	mvninstall	0m 21s	the patch passed
+1	compile	0m 15s	the patch passed with JDK v1.8.0_60
+1	javac	0m 15s	the patch passed

+1	compile	0m 17s	the patch passed with JDK v1.7.0_79
+1	javac	0m 17s	the patch passed
+1	checkstyle	0m 10s	the patch passed
+1	mvneclipse	0m 13s	the patch passed
+1	whitespace	0m 0s	Patch has no whitespace issues.
+1	findbugs	0m 50s	the patch passed
+1	javadoc	0m 14s	the patch passed with JDK v1.8.0_60
+1	javadoc	0m 18s	the patch passed with JDK v1.7.0_79
-1	unit	9m 42s	hadoop-mapreduce-client-app in the patch failed with JDK v1.8.0_60.
-1	unit	10m 9s	hadoop-mapreduce-client-app in the patch failed with JDK v1.7.0_79.
-1	asflicense	0m 22s	Patch generated 7 ASF License warnings.
		29m 45s	

Reason	Tests
JDK v1.7.0_79 Failed junit tests	hadoop.mapreduce.v2.app.TestRecovery
	hadoop.mapreduce.jobhistory.TestJobHistoryEventHandler
	hadoop.mapreduce.jobhistory.TestJobHistoryEventHandler

Subsystem	Report/Notes
Docker	Client=1.7.1 Server=1.7.1 Image:test-patch-base-hadoop-date2015-10-29
JIRA Patch URL	https://issues.apache.org/jira/secure/attachment/12769614/MAPREDUCE-6514.01.patch
JIRA Issue	~~MAPREDUCE-6514~~
Optional Tests	asflicense javac javadoc mvninstall unit findbugs checkstyle compile
uname	Linux 1f4486852554 3.13.0-36-lowlatency #63-Ubuntu SMP PREEMPT Wed Sep 3 21:56:12 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
Build tool	maven
Personality	/home/jenkins/jenkins-slave/workspace/PreCommit-MAPREDUCE-Build/patchprocess/apache-yetus-c3a2069/precommit/personality/hadoop.sh
git revision	trunk / 58d1df5
Default Java	1.7.0_79
Multi-JDK versions	/usr/lib/jvm/java-8-oracle:1.8.0_60 /usr/lib/jvm/java-7-openjdk-amd64:1.7.0_79
findbugs	v3.0.0
unit	https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/6099/artifact/patchprocess/patch-unit-hadoop-mapreduce-project_hadoop-mapreduce-client_hadoop-mapreduce-client-app-jdk1.8.0_60.txt
unit	https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/6099/artifact/patchprocess/patch-unit-hadoop-mapreduce-project_hadoop-mapreduce-client_hadoop-mapreduce-client-app-jdk1.7.0_79.txt
unit test logs	https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/6099/artifact/patchprocess/patch-unit-hadoop-mapreduce-project_hadoop-mapreduce-client_hadoop-mapreduce-client-app-jdk1.8.0_60.txt https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/6099/artifact/patchprocess/patch-unit-hadoop-mapreduce-project_hadoop-mapreduce-client_hadoop-mapreduce-client-app-jdk1.7.0_79.txt
JDK v1.7.0_79 Test Results	https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/6099/testReport/
asflicense	https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/6099/artifact/patchprocess/patch-asflicense-

	problems.txt
modules	C: hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-app U: hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-app
Max memory used	226MB
Powered by	Apache Yetus http://yetus.apache.org
Console output	https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/6099/console

This message was automatically generated.

Varun Saxena added a comment - 30/Oct/15 16:07

Test failure to be handled by ~~YARN-4320~~

Vinod Kumar Vavilapalli added a comment - 30/Oct/15 23:42

Comment on current patch

You should look at `rampDownReduces()` API and use it instead of hand-rolling `decContainerReq`. I actually think once we do this, you should remove `clearAllPendingReduceRequests()` altogether.

I am looking at branch-2 and I think the current patch is better served on top of ~~MAPREDUCE-6302~~ (and this only in 2.8+) given the numerous changes made there. The patch obviously doesn't apply on branch-2.7 which you set the target-version as (2.7.2). Canceling the patch.

Meta thought

If ~~MAPREDUCE-6513~~ goes through per my latest proposal there, there is no need for canceling all the reduce asks and thus this patch, no?

Release

IAC, this has been a long-standing problem (though I'm very surprised nobody caught this till now), so I'd propose we move this out into 2.7.3 or better 2.8+ so I can make progress on the 2.7.2 release. Thoughts?

Wangda Tan added a comment - 05/May/16 18:25

Rebased patch to latest trunk, and addressed comments from vinodkv:

Use rampDownReduces to cancel all reducer requests.

Hadoop QA added a comment - 05/May/16 19:00

✓ +1 overall

Vote	Subsystem	Runtime	Comment
0	reexec	0m 11s	Docker mode activated.
+1	@author	0m 0s	The patch does not contain any @author tags.
+1	test4tests	0m 0s	The patch appears to include 1 new or modified test files.
+1	mvninstall	7m 7s	trunk passed
+1	compile	0m 20s	trunk passed with JDK v1.8.0_91
+1	compile	0m 22s	trunk passed with JDK v1.7.0_95
+1	checkstyle	0m 17s	trunk passed
+1	mvnsite	0m 28s	trunk passed
+1	mvneclipse	0m 13s	trunk passed
+1	findbugs	0m 46s	trunk passed
+1	javadoc	0m 15s	trunk passed with JDK v1.8.0_91
+1	javadoc	0m 18s	trunk passed with JDK v1.7.0_95

+1	mvninstall	0m 23s	the patch passed
+1	compile	0m 17s	the patch passed with JDK v1.8.0_91
+1	javac	0m 17s	the patch passed
+1	compile	0m 20s	the patch passed with JDK v1.7.0_95
+1	javac	0m 20s	the patch passed
+1	checkstyle	0m 14s	the patch passed
+1	mvnsite	0m 25s	the patch passed
+1	mvneclipse	0m 11s	the patch passed
+1	whitespace	0m 0s	Patch has no whitespace issues.
+1	findbugs	0m 53s	the patch passed
+1	javadoc	0m 13s	the patch passed with JDK v1.8.0_91
+1	javadoc	0m 15s	the patch passed with JDK v1.7.0_95
+1	unit	8m 13s	hadoop-mapreduce-client-app in the patch passed with JDK v1.8.0_91.
+1	unit	8m 49s	hadoop-mapreduce-client-app in the patch passed with JDK v1.7.0_95.
+1	asflicense	0m 21s	Patch does not generate ASF License warnings.
		31m 47s	

Subsystem	Report/Notes
Docker	Image:yetus/hadoop:cf2ee45
JIRA Patch URL	https://issues.apache.org/jira/secure/attachment/12802482/MAPREDUCE-6514.02.patch
JIRA Issue	~~MAPREDUCE-6514~~
Optional Tests	asflicense compile javac javadoc mvninstall mvnsite unit findbugs checkstyle
uname	Linux 134bb84d3e93 3.13.0-36-lowlatency #63-Ubuntu SMP PREEMPT Wed Sep 3 21:56:12 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
Build tool	maven
Personality	/testptch/hadoop/patchprocess/precommit/personality/provided.sh
git revision	trunk / d0da132
Default Java	1.7.0_95
Multi-JDK versions	/usr/lib/jvm/java-8-oracle:1.8.0_91 /usr/lib/jvm/java-7-openjdk-amd64:1.7.0_95
findbugs	v3.0.0
JDK v1.7.0_95 Test Results	https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/6483/testReport/
modules	C: hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-app U: hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-app
Console output	https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/6483/console
Powered by	Apache Yetus 0.2.0 http://yetus.apache.org

This message was automatically generated.

Vinod Kumar Vavilapalli added a comment - 05/May/16 23:57

The patch looks good to me now. Tx leftnoteasy for finishing the patch on behalf of varun_saxena!

BTW, why is this a blocker on the 2.6 / 2.7 maint lines?

Wangda Tan added a comment - 06/May/16 00:08

vinodkv,

This behavior is commented by rohithsharma above in above comment,

> As a result, number of containers count in the ask is increased as explained in the below...

And because of ~~MAPREDUCE-6302~~, it is possible that MR AM cancel all reducer requests and re-add all reducer requests at the same heartbeat, so the #containers increases fast in RM side and finally it becomes a enormous number. Since ~~MAPREDUCE-6302~~ is included by branch-2.6/2.7, we need to back port this patch to branch-2.6/2.7 as well.

Hudson added a comment - 06/May/16 02:16

FAILURE: Integrated in Hadoop-trunk-Commit #9728 (See https://builds.apache.org/job/Hadoop-trunk-Commit/9728/)
~~MAPREDUCE-6514~~. Fixed MapReduce ApplicationMaster to properly updated (vinodkv: rev 8d48266720dcf0e71cfd87fef18b60a53aa1bef9)

- hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-app/src/main/java/org/apache/hadoop/mapreduce/v2/app/rm/RMContainerRequestor.java
- hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-app/src/test/java/org/apache/hadoop/mapreduce/v2/app/rm/TestRMContainerAllocator.java
- hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-app/src/main/java/org/apache/hadoop/mapreduce/v2/app/rm/RMContainerAllocator.java

Vinod Kumar Vavilapalli added a comment - 06/May/16 03:47

Okay, makes sense.

The 2.7 and 2.6 commits ran into conflicts and needed trivial fixes - I did them during commit time.

Committed this to trunk, branch-2, branch-2.8, branch-2.7 and branch-2.6. Thanks varun_saxena and leftnoteasy!

Varun Saxena added a comment - 06/May/16 05:49

Thanks leftnoteasy.
Sorry could not address the review comment from Vinod yesterday as it was late night here and waiting for QA report would have meant an even longer wait.

Vinod Kumar Vavilapalli added a comment - 25/Aug/16 22:55

Closing the JIRA as part of 2.7.3 release.

vinisha added a comment - 03/Dec/16 21:16

can you please add affect version to the Jira.

People

Assignee:

Varun Saxena

Reporter:

Varun Saxena

Votes:

0 Vote for this issue

Watchers:

16 Start watching this issue

Dates

Created:

17/Oct/15 06:22

Updated:

06/Jan/17 00:43

Resolved:

06/May/16 03:47