

git_comments:

1. If q is the root queue and the system-wide preemption switch is turned on, then q does not have preemption disabled (default=false, below) unless the `preemption_disabled` property is explicitly set.
2. If the system-wide preemption switch is turned off, all of the queues in the `qPath` hierarchy have preemption disabled, so return true.
3. ****** The specified queue is preemptable if system-wide preemption is turned on ***** unless any queue in the `qPath` hierarchy has explicitly turned ***** preemption off. ***** NOTE: Preemptability is inherited from a queue's parent. ****** @return true if queue has preemption disabled, false otherwise
4. If this is not the root queue, inherit the default value for the `preemption_disabled` property from the parent. Preemptability will be inherited from the parent's hierarchy unless explicitly overridden at this level.
5. ****** Check whether `disable_preemption` property is set for this queue ***** @return true if `disable_preemption` is set, false if not
6. ****** Indicates whether preemption is disabled on the specified queue. ****** @param queue queue path to query ***** @param defaultVal used as default if the `disable_preemption` ***** is not set in the configuration ***** @return true if preemption is disabled on `queue`, false otherwise
7. ****** Sets the `disable_preemption` property in order to indicate ***** whether or not container preemption will be disabled for the specified ***** queue. ****** @param queue queue path ***** @param preemptionDisabled true if preemption is disabled on queue
8. Turn off preemption for queueB and it's children
9. Determine if any of the elements in the `queuepath` have preemption disabled. Also must handle the case where preemption disabled property is explicitly set to something other than the default. Assumes system-wide preemption property is true.
10. Turn off preemption for `queueB(appA)`
11. Turn off preemption for two of the 3 queues with over-capacity.
12. Turn off preemption for `queueA` and it's children. `queueF(appC)`'s request
13. Since `queueB` is not preemptable, get resources from `queueC` Since `queueB` is preemptable, resources will be preempted from both `queueB` and `queueC`. Test must be reset so that the `mDisp`
14. When preemption is turned on for `root.b.b2`, it should be preemptable even though preemption is disabled on `root.b` (parent).
15. Disable preemption at the root queue level. The preemption property should be inherited from root all the way down so that `root.b.b2` should NOT be preemptable.
16. Enable preemption for root (grandparent) but disable for `root.b` (parent). `root.b.b2` should inherit property from parent and NOT be preemptable
17. When preemption turned on for the whole system (`yarn.resourceManager.scheduler.monitor.enable=true`), and with no other preemption properties set, `queue root.b.b2` should be preemptable.

git_commits:

1. **summary:** YARN-2932. Add entry for preemptable status (enabled/disabled) to scheduler web UI and queue initialize/refresh logging. (Eric Payne via wangda)
message: YARN-2932. Add entry for preemptable status (enabled/disabled) to scheduler web UI and queue initialize/refresh logging. (Eric Payne via wangda)

github_issues:**github_issues_comments:****github_pulls:****github_pulls_comments:****github_pulls_reviews:****jira_issues:**

1. **summary:** Add entry for "preemptable" status (enabled/disabled) to scheduler web UI and queue initialize/refresh logging
description: YARN-2056 enables the ability to turn preemption on or off on a per-queue level. This JIRA will provide the preemption status for each queue in the `{{HOST:8088/cluster/scheduler}}` UI and in the RM log during startup/queue refresh.

jira_issues_comments:

1. **body:** Thanks for raising this up [~epayne], it is a good adding. IIRC, YARN-2056 putting the disable preemption configuration for queue code in `ProportionalCapacityPreemptionPolicy` instead of putting them into `CapacitySchedulerConfiguration`. But after read this proposal, I think we should move them to `CapacitySchedulerConfiguration`, and `getIsPreemptionDisabled` should be a method of `CSQueue` interface, thoughts?
label: code-design
2. {quote} IIRC, YARN-2056 putting the disable preemption configuration for queue code in `ProportionalCapacityPreemptionPolicy` instead of putting them into `CapacitySchedulerConfiguration`. But after read this proposal, I think we should move them to `CapacitySchedulerConfiguration`, and `getIsPreemptionDisabled` should be a method of `CSQueue` interface, thoughts? {quote} [~leftnoteasy], yes, I agree. These are good ideas.
3. Hi [~leftnoteasy]. bq. `{{getIsPreemptionDisabled}}` should be a method of `CSQueue` Actually, rather than getting the queue's 'preemption-disable' status, I think it would make more sense to get the queue's preemption status. So, something like `{{getPreemptionStatus}}`. It would return true or false, depending on if queue is preemptable or not. What do you think?
4. bq. Actually, rather than getting the queue's 'preemption-disable' status, I think it would make more sense to get the queue's preemption status. So, something like `getPreemptionStatus`. It would return true or false, depending on if queue is preemptable or not. What do you think? Make sense to me.
5. [~leftnoteasy], thanks for your comments. Would you have some time to take a look at the patch? Thanks, Eric
6. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12687837/YARN-2932.v1.txt> against trunk revision f2d150e. {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 2 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. There were no new javadoc warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:red}-1 findbugs{color}. The patch appears to introduce 14 new Findbugs (version 2.0.3) warnings. {color:green}+1 release

audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager:
org.apache.hadoop.yarn.server.resourcemanager.scheduler.fair.TestAllocationFileLoaderService
org.apache.hadoop.yarn.server.resourcemanager.TestMoveApplication Test results: <https://builds.apache.org/job/PreCommit-YARN-Build/6135/testReport/> Findbugs warnings: <https://builds.apache.org/job/PreCommit-YARN-Build/6135/artifact/patchprocess/newPatchFindbugsWarningshadoop-yarn-server-resourcemanager.html> Console output: <https://builds.apache.org/job/PreCommit-YARN-Build/6135/console> This message is automatically generated.

7. **body:** Hi [~epayne], Thanks for working on this patch, sorry for late response since I was on vacation in the past few weeks. Just took a look at your patch, some comments: 1) Since the `QUEUE_PREEMPTION_DISABLED` is an option for CS, I suggest to make it as a member of `CapacitySchedulerConfiguration`, like `{@getLimitFactor/setLimit}`, etc. This will void some String operations. 2) Rename `{context}` in `{AbstractCSQueue}` to name like `{csContext}` since we have `{rmContext}` 3) I suggest to add a member var like `preemptable` to `AbstractCSQueue`, instead of calling: `{code} + @Private + public boolean isPreemptable() { + return context.getConfiguration().isPreemptable(getQueuePath()); + } {code}` The implementation of `CSConfiguration.isPreemptable(..)` seems too complex to me. `CSConfiguration` should only care about value of configuration file, such logic should put to `AbstractCSQueue.setupQueueConfigs(...)` 4) It's better to web UI name (`preemptable`) and configuration name (`disable_preemption`) consistent. I prefer "preemptable" personally. 5) `{testIsPreemptable}` should be a part of `TestCapacityScheduler` instead of putting it to `TestProportionalCapacityPreemptionPolicy`. 6) In `ProportionalCapacityPreemptionPolicy.cloneQueues`, `preemptable` field should get from Queue instead of getting from configuration. Please let me know your thoughts, Wangda

label: code-design

8. **body:** Thanks very much, [~leftnoteasy], for your thorough review of this patch and for your helpful comments. {quote} 1) Since the `QUEUE_PREEMPTION_DISABLED` is an option for CS, I suggest to make it as a member of `CapacitySchedulerConfiguration`, like `{@getLimitFactor/setLimit}`, etc. This will void some String operations. {quote} This is a good idea. I added `{isQueuePreemptable}` and `{setQueuePreemptable}`. For `{isQueuePreemptable}`, I needed to add a default value parameter because the default for the queue at a particular level should be whatever its parent's value is. {quote} 2) Rename `{context}` in `{AbstractCSQueue}` to name like `{csContext}` since we have `{rmContext}` {quote} Renamed. {quote} 3) I suggest to add a member var like `{preemptable}` to `{AbstractCSQueue}`, instead of calling: `{code} + @Private + public boolean isPreemptable() { + return context.getConfiguration().isPreemptable(getQueuePath()); + } {code}` The implementation of `{CSConfiguration.isPreemptable(..)}` seems too complex to me. `{CSConfiguration}` should only care about value of configuration file, such logic should put to `{AbstractCSQueue.setupQueueConfigs(...)}` {quote} I moved the logic to `{AbstractCSQueue.setupQueueConfigs(...)}`, and you are right. It is much cleaner that way. Thanks! {quote} 4) It's better to web UI name (`preemptable`) and configuration name (`disable_preemption`) consistent. I prefer "preemptable" personally. {quote} Yes, it is less confusing that way. In this patch, the only things that worry about the `{disable_preemption}` property are the internals of the `{CSConfiguration}` methods. The APIs are now all asking whether or not the queue is preemptable. {quote} 5) `{testIsPreemptable}` should be a part of `{TestCapacityScheduler}` instead of putting it to `{TestProportionalCapacityPreemptionPolicy}`. {quote} Thanks. I moved the test to `{testIsPreemptable}`. However, since the interface for changing a queue's preemptability changed, there were also several changes to `{TestProportionalCapacityPreemptionPolicy}`. {quote} 6) In `{ProportionalCapacityPreemptionPolicy.cloneQueues}`, `preemptable` field should get from Queue instead of getting from configuration. {quote} Done.

label: code-design

9. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12691517/YARN-2932.v2.txt> against trunk revision ef3c3a8. {color:red}-1 patch{color}. The patch command could not apply the patch. Console output: <https://builds.apache.org/job/PreCommit-YARN-Build/6303/console> This message is automatically generated.

10. Upmerged and uploading new patch (v3).

11. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12691699/YARN-2932.v3.txt> against trunk revision ae7bf31. {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 3 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:red}-1 javadoc{color}. The javadoc tool appears to have generated 1 warning messages. See <https://builds.apache.org/job/PreCommit-YARN-Build/6312/artifact/patchprocess/diffJavadocWarnings.txt> for details. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 2.0.3) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager:
org.apache.hadoop.yarn.server.resourcemanager.TestWorkPreservingRMRestart Test results: <https://builds.apache.org/job/PreCommit-YARN-Build/6312/testReport/> Console output: <https://builds.apache.org/job/PreCommit-YARN-Build/6312/console> This message is automatically generated.

12. **body:** 1. Rename `isQueuePreemptable` to `getQueuePreemptable` for getter/setter consistency in `CapacitySchedulerConfiguration` 2. Should consider queue reinitialize when queue preemptable in configuration changes (See `{TestQueueParsing}`). And it's best to add a test for verify that. 3. It's better to remove the defaultVal parameter in `CapacitySchedulerConfiguration.isPreemptable`: `{code} public boolean isQueuePreemptable(String queue, boolean defaultVal) {code}` And the default_value should be placed in `CapacitySchedulerConfiguration`, like other queue configuration options. I understand what you trying to do is moving some logic from queue to `CapacitySchedulerConfiguration`, but I still think it's better to keep the `CapacitySchedulerConfiguration` simply gets some values from configuration file. Thanks,

label: code-design

13. [~leftnoteasy], thanks very much for your review and comments: bq. 1. Rename `{isQueuePreemptable}` to `{getQueuePreemptable}` for getter/setter consistency in `{CapacitySchedulerConfiguration}` Renamed. bq. 2. Should consider queue reinitialize when queue preemptable in configuration changes (See `{TestQueueParsing}`). And it's best to add a test for verify that. I'm sorry. I don't understand what you mean by the use of the word "consider." Calling `{CapacityScheduler.reinitialize}` will follow the queue hierarchy down and eventually call `{AbstractCSQueue#setupQueueConfigs}` for every queue, so I don't think there is any additional code needed, unless I'm missing something. Were you just saying that I need to add a test case for that? {quote} 3. It's better to remove the `{defaultVal}` parameter in `{CapacitySchedulerConfiguration.isPreemptable}`: `{code} public boolean isQueuePreemptable(String queue, boolean defaultVal) {code}` And the default_value should be placed in `{CapacitySchedulerConfiguration}`, like other queue configuration options. I understand what you trying to do is moving some logic from queue to `{CapacitySchedulerConfiguration}`, but I still think it's better to keep the `{CapacitySchedulerConfiguration}` simply gets some values from configuration file. {quote} The problem is that without the `{defaultVal}` parameter, `{AbstractCSQueue#isQueuePathHierarchyPreemptable}` can't tell if the queue has explicitly set its preemptability or if it is just returning the default. For example: `{code} root: disable_preemption = true root.A: disable_preemption (the property is not set) root.B: disable_preemption = false (the property is explicitly set to false) {code}` Let's say the `{getQueuePreemptable}` interface is changed to remove the `{defaultVal}` parameter, and that when `{getQueuePreemptable}` calls `{getBoolean}`, it uses `{false}` as the default. # `{getQueuePreemptable}` calls `{getBoolean}` on `{root}` ## `{getBoolean}` returns `{true}` because the `{disable_preemption}` property is set to `{true}` ## `{getQueuePreemptable}` inverts `{true}` and returns `{false}` (That is, `{root}` has preemption disabled, so it is not preemptable). # `{getQueuePreemptable}` calls `{getBoolean}` on `{root.A}` ## `{getBoolean}` returns `{false}` because there is no

- {{disable_preemption}} property set for this queue, so {{getBoolean}} returns the default. ## {{getQueuePreemptable}} inverts {{false}} and returns {{true}} # {{getQueuePreemptable}} calls {{getBoolean}} on {{root.B}} ## {{getBoolean}} returns {{false}} because {{disable_preemption}} property is set to {{false}} for this queue ## {{getQueuePreemptable}} inverts {{false}} and returns {{true}} At this point, {{isQueuePathHierarchyPreemptable}} needs to know if it should use the default preemption from {{root}} or if it should use the value from each child queue. In the case of {{root.A}}, the value from {{root}} ({{false}}) should be used because {{root.A}} does not have the property set. In the case of {{root.B}}, the value should be the one returned for {{root.B}} ({{true}}) because it is explicitly set. But, since both {{root.A}} and {{root.B}} both returned {{true}}, {{isQueuePathHierarchyPreemptable}} can't tell the difference. Does that make sense?
14. [~epayne], Thanks for response, *Re 2:* You're partially correct, queue finally calls setupQueueConfig when reinitialize is invoked. The CapacityScheduler reinitialization is creating a new set of queues, and copy new parameters to your old queues via {code} setupQueueConfigs(clusterResource, newlyParsedLeafQueue.capacity, newlyParsedLeafQueue.absoluteCapacity, newlyParsedLeafQueue.maximumCapacity, newlyParsedLeafQueue.absoluteMaxCapacity, ... {code} So you need put the parameter you wants to update to setupQueueConfig as well. Without that, queue will not be refreshed. I didn't find any changes to parameter of setupQueueConfig, so I guess so, it's better to add a test to verify it. *Re 3:* You can take a look at how AbstractCSQueue initialize labels, {code} // get labels this.accessibleLabels = cs.getConfiguration().getAccessibleNodeLabels(getQueuePath()); // inherit from parent if labels not set if (this.accessibleLabels == null && parent != null) { this.accessibleLabels = parent.getAccessibleNodeLabels(); } {code} I think they have similar logic -- For node label is trying to get value from configuration, if not set, inherit from parent. With this, you can make getPreemptable interface without defaultVal in CapacitySchedulerConfiguration.
15. Since YARN-2056 is also planned in 2.7 (I thought it's already included in 2.6), do you think is it better to make configuration option name to <queue-patch>.preemptable for consistency?
16. **body:** Updated title of the JIRA a little bit to reflect what we have done in the patch.
label: code-design
17. Thank you [~leftnoteasy] for your review and comments. {quote} Re 2: You're partially correct, queue finally calls setupQueueConfig when reinitialize is invoked. The CapacityScheduler reinitialization is creating a new set of queues, and copy new parameters to your old queues via {code} setupQueueConfigs(clusterResource, newlyParsedLeafQueue.capacity, ... {code} So you need put the parameter you wants to update to setupQueueConfig as well. Without that, queue will not be refreshed. I didn't find any changes to parameter of setupQueueConfig, so I guess so, it's better to add a test to verify it. {quote} I made the changes to the API for {{AbstractCSQueue#setupQueueConfigs}} to take the additional preemptable parameter. When it is called from {{[Leaf|Parent]Queue#setupQueueConfigs}}, it calls {{AbstractCSQueue#isQueuePathHierarchyPreemptable}} to get the preemptability of the queue. I tested the fixes in both version 3 and version 4 of this patch on a one-node cluster and on a 10-node cluster. In both version, I was able to change the {{disable_preemption}} properties, refresh the queues using {{yarn rmadmin -refreshQueues}}, and I was able to see the updates on the Scheduler UI page. However, I think I see that if the new list of queues is different than the old list of queues, it would not pick up the parameters for the new queues without this change. {quote} Re 3: You can take a look at how AbstractCSQueue initialize labels, I think they have similar logic -- For node label is trying to get value from configuration, if not set, inherit from parent. With this, you can make getPreemptable interface without defaultVal in CapacitySchedulerConfiguration. {quote} I did change {{CapacitySchedulerConfiguration#getQueuePreemptable}} to not take a default value, but in order to pass back the {{null}} information, it has to return a {{String}} and then the caller has to convert the {{String}} to a Boolean, which I think is a little awkward. {quote} Since YARN-2056 is also planned in 2.7 (I thought it's already included in 2.6), do you think is it better to make configuration option name to <queue-patch>.preemptable for consistency? {quote} Well, that would be ideal, I think, but it isn't that simple on our side. We have already started using the code in YARN-2056 and are using the {{disable_preemption}} property. An argument could be made that {{disable_preemption}} is better because it indicates that it is turning off the {{...monitor.capacity.preemption...}} property. If {{disable_preemption}} were changed to {{preemptable}}, someone may look at that property and think that the queue should have that property without considering the overall, system property {{...monitor.capacity.preemption...}}. How important is it to you that {{disable_preemption}} property be changed to {{preemptable}}?
18. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12692570/YARN-2932.v4.txt> against trunk revision 9e33116. {color:red}-1 patch{color}. The patch command could not apply the patch. Console output: <https://builds.apache.org/job/PreCommit-YARN-Build/6344/console> This message is automatically generated.
19. Uploading patch v5. Should apply, now, to both branch-2 and trunk.
20. **body:** [~epayne], Thanks for update, after looked at your latest patch, I think ver.3 is more clear to me comparing to ver.5. It looks bad to check the {{null}} at {{getQueuePreemptable}}. And ver.3 supports queue reinitialize, it was my misunderstanding about logics in your patch, sorry for adding unnecessary work to you. Two nits on ver.3: 1) In {{isQueuePathHierarchyPreemptable}} You can use: parentQ.preemptable instead of isQueuePathHierarchyPreemptable(parentQ). Instead of {code} + return csConf.isQueuePreemptable(q.getQueuePath(), + isQueuePathHierarchyPreemptable(parentQ)); {code} The latter one is a recursive method call and potentially makes debug harder. CS.parseQueue assumes parent queue get initialized before children queues. 2) Still in {{isQueuePathHierarchyPreemptable}} Use {{DEFAULT_RM_SCHEDULER_ENABLE_MONITORS}} instead of hard coded {{false}} when getting system-wide-preemptable. It is possible in the future we turn on preemption policy by default, only one place we will need to edit in that case. Besides the two nits, patch LGTM, could you rebase ver.3 against latest trunk and kick Jenkins? Thanks, Wangda
label: code-design
21. Thank you, [~leftnoteasy], for your help and advice on this patch. I have made the changes you suggested to v6 of the patch.
22. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12693664/YARN-2932.v6.txt> against trunk revision 4271554. {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 3 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:red}-1 javadoc{color}. The javadoc tool appears to have generated 1 warning messages. See <https://builds.apache.org/job/PreCommit-YARN-Build/6378/artifact/patchprocess/diffJavadocWarnings.txt> for details. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 2.0.3) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:green}+1 core tests{color}. The patch passed unit tests in hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager. Test results: <https://builds.apache.org/job/PreCommit-YARN-Build/6378/testReport/> Console output: <https://builds.apache.org/job/PreCommit-YARN-Build/6378/console> This message is automatically generated.
23. **body:** Version 7 of patch fixes new javadoc warnings. Sorry about that.
label: documentation
24. **body:** Hi [~epayne], Thanks for update, I've just tried run a cluster with this patch. I feel we'd better to make config and web UI consistent, the difference between config and web UI a little confusing. If it is not easy to modify config, I suggest to make CapacitySchedulerPage shows "preemption disabled" and also logging in LeafQueue. Otherwise, patch LGTM. Wangda
label: code-design
25. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12693705/YARN-2932.v7.txt> against trunk revision c0af72c. {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 3 new or modified test files.

- {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. There were no new javadoc warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 2.0.3) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager: org.apache.hadoop.yarn.server.resourcemanager.applicationsmanager.TestAMRestart org.apache.hadoop.yarn.server.resourcemanager.recovery.TestFSRMStateStore org.apache.hadoop.yarn.server.resourcemanager.scheduler.fair.TestFairScheduler Test results: <https://builds.apache.org/job/PreCommit-YARN-Build/6379/testReport/> Console output: <https://builds.apache.org/job/PreCommit-YARN-Build/6379/console> This message is automatically generated.
26. Thanks, [~leftnoteasy], for your reviews and your help working through this jira. {quote} I feel we'd better to make config and web UI consistent, the difference between config and web UI a little confusing. If it is not easy to modify config, I suggest to make CapacitySchedulerPage shows "preemption disabled" and also logging in LeafQueue. {quote} I'm afraid I couldn't make it all "isPreemptable," so I made everything reflect the state of "disable_preemption". It's actually not too bad. I'll upload a screenshot.
27. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12694392/Screenshot.Queue.Preemption.Disabled.jpg> against trunk revision 8f26d5a. {color:red}-1 patch{color}. The patch command could not apply the patch. Console output: <https://builds.apache.org/job/PreCommit-YARN-Build/6411/console> This message is automatically generated.
28. {color:green}+1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12694391/YARN-2932.v8.txt> against trunk revision 8f26d5a. {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 3 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. There were no new javadoc warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 2.0.3) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:green}+1 core tests{color}. The patch passed unit tests in hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager. Test results: <https://builds.apache.org/job/PreCommit-YARN-Build/6410/testReport/> Console output: <https://builds.apache.org/job/PreCommit-YARN-Build/6410/console> This message is automatically generated.
29. Thanks for updating, [~epayne]. New patch & screenshot looks good to me, will commit it tomorrow if no objections.
30. Just committed to trunk/branch-2, thanks [~epayne] reporting and working on it!
31. SUCCESS: Integrated in Hadoop-trunk-Commit #6946 (See [<https://builds.apache.org/job/Hadoop-trunk-Commit/6946/>]) YARN-2932. Add entry for preemptable status (enabled/disabled) to scheduler web UI and queue initialize/refresh logging. (Eric Payne via wangda) (wangda: rev 18741adf97f4fda5f8743318b59c440928e51297) * hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/webapp/CapacitySchedulerPage.java * hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/webapp/dao/CapacitySchedulerLeafQueueInfo.java * hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/scheduler/capacity/AbstractCSQueue.java * hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/scheduler/capacity/LeafQueue.java * hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager/src/test/java/org/apache/hadoop/yarn/server/resourcemanager/webapp/TestRMWebServicesCapacitySched.java * hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager/src/test/java/org/apache/hadoop/yarn/server/resourcemanager/monitor/capacity/TestProportionalCapacityPreemptionPolicy.java * hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/scheduler/capacity/CapacitySchedulerConfiguration.java * hadoop-yarn-project/CHANGES.txt * hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/scheduler/capacity/CSQueue.java * hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager/src/test/java/org/apache/hadoop/yarn/server/resourcemanager/scheduler/capacity/TestCapacityScheduler.java * hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/monitor/capacity/ProportionalCapacityPreemptionPolicy.java
32. FAILURE: Integrated in Hadoop-Yarn-trunk-Java8 #87 (See [<https://builds.apache.org/job/Hadoop-Yarn-trunk-Java8/87/>]) YARN-2932. Add entry for preemptable status (enabled/disabled) to scheduler web UI and queue initialize/refresh logging. (Eric Payne via wangda) (wangda: rev 18741adf97f4fda5f8743318b59c440928e51297) * hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/monitor/capacity/ProportionalCapacityPreemptionPolicy.java * hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/webapp/CapacitySchedulerPage.java * hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager/src/test/java/org/apache/hadoop/yarn/server/resourcemanager/webapp/TestRMWebServicesCapacitySched.java * hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager/src/test/java/org/apache/hadoop/yarn/server/resourcemanager/monitor/capacity/TestProportionalCapacityPreemptionPolicy.java * hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/scheduler/capacity/AbstractCSQueue.java * hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/scheduler/capacity/CSQueue.java * hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/webapp/dao/CapacitySchedulerLeafQueueInfo.java * hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager/src/test/java/org/apache/hadoop/yarn/server/resourcemanager/scheduler/capacity/TestCapacityScheduler.java * hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/scheduler/capacity/LeafQueue.java * hadoop-yarn-project/CHANGES.txt * hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/scheduler/capacity/CapacitySchedulerConfiguration.java
33. FAILURE: Integrated in Hadoop-Yarn-trunk #821 (See [<https://builds.apache.org/job/Hadoop-Yarn-trunk/821/>]) YARN-2932. Add entry for preemptable status (enabled/disabled) to scheduler web UI and queue initialize/refresh logging. (Eric Payne via wangda) (wangda: rev 18741adf97f4fda5f8743318b59c440928e51297) * hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/webapp/dao/CapacitySchedulerLeafQueueInfo.java * hadoop-

[illegible]

project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-
resourcemanager/src/test/java/org/apache/hadoop/yarn/server/resourcemanager/webapp/TestRMWebServicesCapacitySched.java * hadoop-
yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-
resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/scheduler/capacity/AbstractCSQueue.java * hadoop-yarn-
project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-
resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/scheduler/capacity/LeafQueue.java * hadoop-yarn-
project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-
resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/monitor/capacity/ProportionalCapacityPreemptionPolicy.java

38. FAILURE: Integrated in Hadoop-Mapreduce-trunk #2038 (See [<https://builds.apache.org/job/Hadoop-Mapreduce-trunk/2038/>]) YARN-2932.
Add entry for preemptable status (enabled/disabled) to scheduler web UI and queue initialize/refresh logging. (Eric Payne via wangda)
(wangda: rev 18741adf97f4fda5f8743318b59c440928e51297) * hadoop-yarn-project/CHANGES.txt * hadoop-yarn-project/hadoop-
yarn/hadoop-yarn-server/hadoop-yarn-server-
resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/scheduler/capacity/CapacitySchedulerConfiguration.java *
hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-
resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/scheduler/capacity/CSQueue.java * hadoop-yarn-
project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-
resourcemanager/src/test/java/org/apache/hadoop/yarn/server/resourcemanager/scheduler/capacity/TestCapacityScheduler.java * hadoop-yarn-
project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-
resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/webapp/CapacitySchedulerPage.java * hadoop-yarn-
project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-
resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/webapp/dao/CapacitySchedulerLeafQueueInfo.java * hadoop-
yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-
resourcemanager/src/test/java/org/apache/hadoop/yarn/server/resourcemanager/webapp/TestRMWebServicesCapacitySched.java * hadoop-
yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-
resourcemanager/src/test/java/org/apache/hadoop/yarn/server/resourcemanager/monitor/capacity/TestProportionalCapacityPreemptionPolicy.java
* hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-
resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/scheduler/capacity/LeafQueue.java * hadoop-yarn-
project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-
resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/monitor/capacity/ProportionalCapacityPreemptionPolicy.java
* hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-
resourcemanager/src/main/java/org/apache/hadoop/yarn/server/resourcemanager/scheduler/capacity/AbstractCSQueue.java