Item 187

git_comments:

1. * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
2. disable renewing leases as this will force spooling to happen.

git_commits:

1. **summary:** PHOENIX-5909 - Table and index-level metrics for indexing coprocs
   message: PHOENIX-5909 - Table and index-level metrics for indexing coprocs

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Table and index-level metrics for indexing coprocs
   description: For many of the metrics which the IndexRegionObserver and GlobalIndexChecker coprocs expose, it would be useful to have them broken down by table or index, in addition to the current aggregates by region server. This would help us determine if, for example, a flood of read repairs was coming from one particular index. Thanks [~priyankporwal] for the feature suggestion.

jira_issues_comments:

1. [~ChinmayKulkarni], [~priyankporwal], [~kadir], [~swaroopa]: What constitutes an acceptable test here? PHOENIX-4042 (global indexing metrics for Indexer) and PHOENIX-5358 (metrics for GlobalIndexChecker) don't seem to have included tests at all, but I know we're trying to improve our code coverage. The client side metrics have nice tests (e.g PhoenixMetricsIT), but I don't see a lot of instrumentation to verify region-server-side metrics. Is a simple unit test of the two MetricsImpl classes sufficient, or does all the server-side instrumentation need to be built so we can verify as part of IT tests as well? Or is that a separate JIRA?
2. {color:red}-1 overall{color}. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/13003786/PHOENIX-5909-4.x-v1.patch against 4.x branch at commit aad583670ea821286ab5e2460ce0ab7255d474c4. ATTACHMENT ID: 13003786 {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 4 new or modified tests. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 lineLengths{color}. The patch introduces the following lines longer than 100: + props.put(QueryServices.TASK_HANDLING_INITIAL_DELAY_MS_ATTRIB, Long.toString(Long.MAX_VALUE)); + verifyCounter(getTableCounterName(MetricsIndexerSource.POST_INDEX_UPDATE_FAILURE), registry); {color:red}-1 core tests{color}. The patch failed these unit tests: ./phoenix-core/target/failsafe-reports/TEST-org.apache.phoenix.end2end.index.LocalImmutableTxIndexIT ./phoenix-core/target/failsafe-reports/TEST-org.apache.phoenix.tx.TxCheckpointIT ./phoenix-core/target/failsafe-reports/TEST-org.apache.phoenix.end2end.index.MutableIndexIT ./phoenix-core/target/failsafe-

reports/TEST-org.apache.phoenix.end2end.join.HashJoinGlobalIndexIT ./phoenix-core/target/failsafe-reports/TEST-org.apache.phoenix.end2end.join.SubqueryUsingSortMergeJoinIT ./phoenix-core/target/failsafe-reports/TEST-org.apache.phoenix.end2end.join.HashJoinMoreIT ./phoenix-core/target/failsafe-reports/TEST-org.apache.phoenix.end2end.salted.SaltedTableVarLengthRowKeyIT Test results: https://builds.apache.org/job/PreCommit-PHOENIX-Build/3889//testReport/ Code Coverage results: https://builds.apache.org/job/PreCommit-PHOENIX-Build/3889//artifact/phoenix-core/target/site/jacoco/index.html Console output: https://builds.apache.org/job/PreCommit-PHOENIX-Build/3889//console This message is automatically generated.

3. This patch was approved back in May but by mistake never merged in. Rebasing on current 4.x and rerunning tests.
4. | (x) *{color:red}-1 overall{color}* | \\ \\ || Vote || Subsystem || Runtime || Comment || | {color:blue}0{color} | {color:blue} reexec {color} | {color:blue} 0m 0s{color} | {color:blue} Docker mode activated. {color} | | {color:red}-1{color} | {color:red} patch {color} | {color:red} 0m 7s{color} | {color:red} PHOENIX-5909 does not apply to master. Rebase required? Wrong Branch? See https://yetus.apache.org/documentation/in-progress/precommit-patchnames for help. {color} | \\ \\ || Subsystem || Report/Notes || | JIRA Issue | PHOENIX-5909 | | JIRA Patch URL | https://issues.apache.org/jira/secure/attachment/13011979/PHOENIX-5909-4.x-v2.patch | | Console output | https://ci-hadoop.apache.org/job/PreCommit-PHOENIX-Build/36/console | | versions | git=2.17.1 | | Powered by | Apache Yetus 0.12.0 https://yetus.apache.org | This message was automatically generated.
5. | (x) *{color:red}-1 overall{color}* | \\ \\ || Vote || Subsystem || Runtime || Comment || | {color:blue}0{color} | {color:blue} reexec {color} | {color:blue} 1m 24s{color} | {color:blue} Docker mode activated. {color} | || || || || {color:brown} Prechecks {color} || | {color:green}+1{color} | {color:green} dupname {color} | {color:green} 0m 0s{color} | {color:green} No case conflicting files found. {color} | | {color:green}+1{color} | {color:green} hbaseanti {color} | {color:green} 0m 1s{color} | {color:green} Patch does not have any anti-patterns. {color} | | {color:green}+1{color} | {color:green} @author {color} | {color:green} 0m 0s{color} | {color:green} The patch does not contain any @author tags. {color} | | {color:red}-1{color} | {color:red} test4tests {color} | {color:red} 0m 0s{color} | {color:red} 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. {color} | || || || || {color:brown} master Compile Tests {color} || | {color:green}+1{color} | {color:green} mvninstall {color} | {color:green} 13m 14s{color} | {color:green} master passed {color} | | {color:green}+1{color} | {color:green} compile {color} | {color:green} 1m 10s{color} | {color:green} master passed {color} | | {color:green}+1{color} | {color:green} checkstyle {color} | {color:green} 0m 56s{color} | {color:green} master passed {color} | | {color:green}+1{color} | {color:green} javadoc {color} | {color:green} 0m 49s{color} | {color:green} master passed {color} | | {color:blue}0{color} | {color:blue} spotbugs {color} | {color:blue} 3m 30s{color} | {color:blue} phoenix-core in master has 973 extant spotbugs warnings. {color} | || || || || {color:brown} Patch Compile Tests {color} || | {color:red}-1{color} | {color:red} mvninstall {color} | {color:red} 1m 14s{color} | {color:red} root in the patch failed. {color} | | {color:green}+1{color} | {color:green} compile {color} | {color:green} 1m 7s{color} | {color:green} the patch passed {color} | | {color:green}+1{color} | {color:green} javac {color} | {color:green} 1m 7s{color} | {color:green} the patch passed {color} | | {color:red}-1{color} | {color:red} checkstyle {color} | {color:red} 1m 0s{color} | {color:red} phoenix-core: The patch generated 92 new + 1085 unchanged - 32 fixed = 1177 total (was 1117) {color} | | {color:green}+1{color} | {color:green} whitespace {color} | {color:green} 0m 0s{color} | {color:green} The patch has no whitespace issues. {color} | | {color:green}+1{color} | {color:green} javadoc {color} | {color:green} 0m 48s{color} | {color:green} the patch passed {color} | | {color:green}+1{color} | {color:green} spotbugs {color} | {color:green} 3m 41s{color} | {color:green} the patch passed {color} | || || || || {color:brown} Other Tests {color} || | {color:red}-1{color} | {color:red} unit {color} | {color:red}108m 37s{color} | {color:red} phoenix-core in the patch failed. {color} | | {color:green}+1{color} | {color:green} asflicense {color} | {color:green} 0m 25s{color} | {color:green} The patch does not generate ASF License warnings. {color} | | {color:black}{color} | {color:black} {color} | {color:black}138m 37s{color} | {color:black} {color} | \\ \\ || Subsystem || Report/Notes || | Docker | ClientAPI=1.40 ServerAPI=1.40 base: https://ci-hadoop.apache.org/job/PreCommit-PHOENIX-Build/37/artifact/patchprocess/Dockerfile | | JIRA Issue | PHOENIX-5909 | | JIRA Patch URL | https://issues.apache.org/jira/secure/attachment/13011980/PHOENIX-5909.master.v1.patch | | Optional Tests | dupname asflicense javac javadoc unit spotbugs hbaseanti checkstyle compile | | uname | Linux 5893fcd664aa 4.15.0-112-generic #113-Ubuntu SMP Thu Jul 9 23:41:39 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux | | Build tool | maven | | Personality | dev/phoenix-personality.sh | | git revision | master / 10738f9 | | Default Java | Private Build-1.8.0_242-8u242-b08-0ubuntu3~16.04-b08 | | mvninstall

| https://ci-hadoop.apache.org/job/PreCommit-PHOENIX-Build/37/artifact/patchprocess/patch-mvninstall-root.txt | | checkstyle | https://ci-hadoop.apache.org/job/PreCommit-PHOENIX-Build/37/artifact/patchprocess/diff-checkstyle-phoenix-core.txt | | unit | https://ci-hadoop.apache.org/job/PreCommit-PHOENIX-Build/37/artifact/patchprocess/patch-unit-phoenix-core.txt | | Test Results | https://ci-hadoop.apache.org/job/PreCommit-PHOENIX-Build/37/testReport/ | | Max. process+thread count | 6472 (vs. ulimit of 30000) | | modules | C: phoenix-core U: phoenix-core | | Console output | https://ci-hadoop.apache.org/job/PreCommit-PHOENIX-Build/37/console | | versions | git=2.7.4 maven=3.3.9 spotbugs=4.1.2 | | Powered by | Apache Yetus 0.12.0 https://yetus.apache.org | This message was automatically generated.

6. This had been approved months ago but due to an oversight on my part had never been committed. I rebased the 4.x patch and ported to master. Test on master passed aside from one (IndexWithSchemaChangeIT) which crashed and passed when rerun locally. Merged to both 4.x and master.