Corpus Details

# Apache HBASE

## Title:

Title: [HBASE-15422] Procedure v2 - Avoid double yield

Title: [HBASE-15421] Convert TestStoreScanner to junit4 from junit3 and clean up close of scanners

Title: [HBASE-15420] TestCacheConfig failed after HBASE-15338

Title: [HBASE-15419] Push HBASE-14850 to apache's git.

Title: [HBASE-15418] Clean up un-used warning in test util

Title: [HBASE-15417] Calls to ObserverContext#bypass in a region observer's prePut method are inconsistent

Title: [HBASE-15416] TestHFileBackedByBucketCache is flakey since it went in

Title: [HBASE-15415] Improve Master WebUI snapshot information

Title: [HBASE-15414] Bound the size of multi request returns and/or allow return of partial result up to client

Title: [HBASE-15413] Procedure-V2: print out ProcedureInfo during trace

Title: [HBASE-15412] Add average region size metric

Title: [HBASE-15411] Rewrite backup with Procedure V2

Title: [HBASE-15410] Utilize the max seek value when all Filters in MUST\_PASS\_ALL FilterList return SEEK\_NEXT\_USING\_HINT

Title: [HBASE-15409] TestHFileBackedByBucketCache failed randomly on jdk8

Title: [HBASE-15408] MiniCluster's master crashes and unit tests timeout

Title: [HBASE-15407] Add SASL support for fan out OutputStream

Title: [HBASE-15406] Split / merge switch left disabled after early termination of hbck

Title: [HBASE-15405] Fix PE logging and wrong defaults in help message

Title: [HBASE-15404] PE: Clients in append and increment are operating serially

Title: [HBASE-15403] Performance Evaluation tool isn't working as expected

Title: [HBASE-15402] Add on thrift server that uses cpp client.

Title: [HBASE-15401] Update third-party deps and add Zoopeeker

Title: [HBASE-15400] Use DateTieredCompactor for Date Tiered Compaction

Title: [HBASE-15399] JavaHBaseContext is missung the BulkLoad calls.

Title: [HBASE-15398] Cells loss or disorder when using family essential filter and partial scanning protocol

Title: [HBASE-15397] Create bulk load replication znode(hfile-refs) in ZK replication queue by default

Title: [HBASE-15396] Enhance mapreduce.TableSplit to add encoded region name

Title: [HBASE-15395] HMaster ui sometimes doesn't display anything under user/system tables

Title: [HBASE-15394] Orphan daughter region directories are not cleared which have been left around from previous split attempt

Title: [HBASE-15393] Enable table replication command will fail when parent znode is not default in peer cluster

Title: [HBASE-15392] Single Cell Get reads two HFileBlocks

Title: [HBASE-15391] Avoid too large "deleted from META" info log

Title: [HBASE-15390] ClientExceptionUtils doesn't handle CallQueueTooBigException properly

Title: [HBASE-15389] Write out multiple files when compaction

Title: [HBASE-15388] Add ACL for some master methods

Title: [HBASE-15387] Make HalfStoreFileReader configurable in LoadIncrementalHFiles

Title: [HBASE-15386] PREFETCH\_BLOCKS\_ON\_OPEN in HColumnDescriptor is ignored

Title: [HBASE-15385] A failed atomic folder rename operation can never recovery for the destination file is deleted in Wasb filesystem

Title: [HBASE-15384] Avoid using '/tmp' directory in TestBulkLoad

Title: [HBASE-15383] Load distribute across secondary read replicas for meta

Title: [HBASE-15382] Expose regionserver metadata (ie groups, tables, servers) via JMX

Title: [HBASE-15381] Implement a distributed MOB compaction by procedure

Title: [HBASE-15380] Purge rollback support in Store etc.

Title: [HBASE-15379] Fake cells created in read path not implementing SettableSequenceId

Title: [HBASE-15378] Scanner can not handle a heartbeat message with no results

Title: [HBASE-15377] Per-RS Get metric is time based, per-region metric is size-based

Title: [HBASE-15376] ScanNext metric is size-based while every other per-operation metric is time based

Title: [HBASE-15375] Do not write to '/tmp' in TestRegionMover

Title: [HBASE-15374] Avoid using '/tmp' directory in our unit tests

Title: [HBASE-15373] DEPRECATED\_NAME\_OF\_NO\_LIMIT\_THROUGHPUT\_CONTROLLER\_CLASS value is wrong in CompactionThroughputControllerFactory

Title: [HBASE-15372] HBase Backup/Restore Phase 2: Keep WAL files in a centralized place to avoid duplication

Title: [HBASE-15371] Procedure V2 - Completed support parent-child procedure

Title: [HBASE-15370] Backport Moderate Object Storage (MOB) to branch-1

Title: [HBASE-15369] Handle NPE in region.jsp

Title: [HBASE-15368] Add relative window support

Title: [HBASE-15367] Configuring DisabledWAL provider hangs the RegionServer

Title: [HBASE-15366] Add doc, trace-level logging, and test around hfileblock

Title: [HBASE-15365] Do not write to '/tmp' in TestHBaseConfiguration

Title: [HBASE-15364] Fix unescaped < characters in Javadoc

Title: [HBASE-15363] Add client side metrics for SASL connection failures

Title: [HBASE-15362] Compression Algorithm does not respect config params from hbase-site

Title: [HBASE-15361] Remove unnecessary or Document constraints on BucketCache possible bucket sizes

Title: [HBASE-15360] Fix flaky TestSimpleRpcScheduler

Title: [HBASE-15359] Simplifying Segment hierarchy

Title: [HBASE-15358] canEnforceTimeLimitFromScope should use timeScope instead of sizeScope

Title: [HBASE-15357] TableInputFormatBase getSplitKey does not handle signed bytes correctly

Title: [HBASE-15356] Remove unused Imports

Title: [HBASE-15355] region.jsp can not be found on info server of master

Title: [HBASE-15354] Use same criteria for clearing meta cache for all operations

Title: [HBASE-15353] Add metric for number of CallQueueTooBigException's

Title: [HBASE-15352] FST BlockEncoder

Title: [HBASE-15351] Fix description of hbase.bucketcache.size in hbase-default.xml

Title: [HBASE-15350] Enable individual unit test in hbase-spark module

Title: [HBASE-15349] Update surefire version to 2.19.1

Title: [HBASE-15348] Fix tests broken by recent metrics re-work

Title: [HBASE-15347] Update CHANGES.txt for 1.3

Title: [HBASE-15346] add 1.3 RM to docs

Title: [HBASE-15345] add branch-1.3 post-commit builds

Title: [HBASE-15344] add 1.3 to prereq tables in ref guide

Title: [HBASE-15343] add branch-1.3 to precommit branches

Title: [HBASE-15342] create branch-1.3 and update branch-1 poms to 1.4.0-SNAPSHOT

Title: [HBASE-15341] 1.3 release umbrella

Title: [HBASE-15340] Partial row result of scan may return data violates the row-level transaction

Title: [HBASE-15339] Improve DateTieredCompactionPolicy

Title: [HBASE-15338] Add a option to disable the data block cache for testing the performance of underlying file system

Title: [HBASE-15337] Document FIFO and date tiered compaction in the book

Title: [HBASE-15336] Support Dataframe writer to the connector

Title: [HBASE-15335] Add composite key support in row key

Title: [HBASE-15334] Add avro support for spark hbase connector

Title: [HBASE-15333] Enhance the filter to handle short, integer, long, float and double

Title: [HBASE-15332] Document how to take advantage of HDFS-6133 in HBase

Title: [HBASE-15331] HBase Backup/Restore Phase 2: Optimized Restore operation

Title: [HBASE-15330] HBase Backup/Restore Phase 3: support delete/truncate table

Title: [HBASE-15329] Cross-Site Scripting: Reflected in table.jsp

Title: [HBASE-15328] Unvalidated Redirect in HMaster

Title: [HBASE-15327] Canary will always invoke admin.balancer() in each sniffing period when writeSniffing is enabled

Title: [HBASE-15326] NPE in TestHRegion.testBatchPut\_whileNoRowLocksHeld

Title: [HBASE-15325] ResultScanner allowing partial result will miss the rest of the row if the region is moved between two rpc requests

Title: [HBASE-15324] Jitter may cause desiredMaxFileSize overflow in ConstantSizeRegionSplitPolicy and trigger unexpected split

Title: [HBASE-15323] Hbase Rest CheckAndDeleteAPi should be able to delete more cells

## Description:

Description: <p>ServerCrashProcedure is using a combination of isYieldBeforeExecuteFromState() and ProcedureYieldException, which may end up in yielding twice. (ServerCrashProcedure is the only user of yield)</p>

Description: <p>I want to add to this test class but it junit3. Let me convert it over first.</p>

Description: <p>TestCacheConfig failed after <a href="https://issues.apache.org/jira/browse/HBASE-15338" title="Add a option to disable the data block cache for testing the performance of underlying file system" class="issue-link" data-issue-key="HBASE-15338"><del>HBASE-15338</del></a>.<br/>

Fix it in this issue~</p>

Description: <p>Now that Apache's git isn't locked down it's time to push the code where it should be.</p>

Description:

Description: <p>Calling ctx.bypass(), where ctx is the ObserverContext object passed in to the region observer's prePut method, results in some inconsistent behavior.</p>

<p>If every other put in the batch is also bypassed, the region observer sees none of these in its postPut method. If there is at least one other put which is not bypassed, the region observer sees all of the puts in the batch <em>including those which were bypassed</em>.</p>

<p>The end result is that, after bypassing a put, that put may or may not end up in the region observer's postPut method. This behavior is dependent solely on which other puts the bypassed put is batched together with.</p>

<p>I tried to find existing tickets for this issue, but was unable to. Apologies if I missed something. The closest issues I could find were <a href="https://issues.apache.org/jira/browse/HBASE-4331" title="Bypassing default actions in prePut fails sometimes with HTable client" class="issue-link" data-issue-key="HBASE-4331"><del>HBASE-4331</del></a> and <a href="https://issues.apache.org/jira/browse/HBASE-11503" title="Inconsistency in CP between HRegion#batchMutate() and MultiRowMutation()" class="issue-link" data-issue-key="HBASE-11503">HBASE-11503</a>, but those didn't seem to quite hit it.</p>

<p>Additionally, I threw together a quick demonstration of this issue: <a href="https://github.com/hwh33/bypass-inconsistency-demo" class="external-link" rel="nofollow">https://github.com/hwh33/bypass-inconsistency-demo</a>. You can run that demo in memory using the testing utility or against a running cluster. I actually haven't had time to test it against a cluster though, so you may encounter bugs if running in that mode (but hopefully not!).</p>

Description: <p>Looks like cache content changes during running of test... let me fix. Critical.</p>

Description: <p>On the Master WebUI, we currently show lots of information about the space used by individual snapshots.</p>

<p>We should also give a total space used.</p>

Description: <p>Some knowledgeable hbase users note that while Scanning now allows you return results in 'chunks' for assembly client-side as a whole result (or the application can see the partials as they come out of the cluster), this ability is absent if you do a multi-get; you might get back more than you bargained for and just as chunking when Scanning makes sense because it makes hbase 'regular', we need the same for multiget.</p>

<p>Parking an issue here for discussion.</p>

Description: <p>Before the <a href="https://issues.apache.org/jira/browse/HBASE-15100" title="Master WALProcs still never clean up" class="issue-link" data-issue-key="HBASE-15100"><del>HBASE-15100</del></a> refactored the code to print the ProcedureInfo object, we don't have the needs. Now we need to implement a customized toString() function to print out information in the ProcedureInfo object.</p>

Description: <p>We have several metrics related to region store file size, num regions, etc per regionserver, but we do not have a single metric to track the average region size per regionserver. </p>

<p>Avg region size is important to look at for deciding on the split policy, etc.</p>

Description: <p>Currently full / incremental backup is driven by BackupHandler (see call() method for flow).</p>

<p>This issue is to rewrite the flow using Procedure V2.</p>

<p>States (enum) for full / incremental backup would be introduced in Backup.proto which correspond to the steps performed in BackupHandler#call().<br/>

executeFromState() would pace the backup based on the current state.<br/>

serializeStateData() / deserializeStateData() would be used to persist state into procedure WAL.</p>

Description: <p>As Preston mentioned in the comment in <a href="https://issues.apache.org/jira/browse/HBASE-15243" title="Utilize the lowest seek value when all Filters in MUST\_PASS\_ONE FilterList return SEEK\_NEXT\_USING\_HINT" class="issue-link" data-issue-key="HBASE-15243"><del>HBASE-15243</del></a>:</p>

<p><a href="https://issues.apache.org/jira/browse/HBASE-15243?focusedCommentId=15143557&amp;page=com.atlassian.jira.plugin.system.issuetabpanels:comment-tabpanel#comment-15143557" class="external-link" rel="nofollow">https://issues.apache.org/jira/browse/HBASE-15243?focusedCommentId=15143557&amp;page=com.atlassian.jira.plugin.system.issuetabpanels:comment-tabpanel#comment-15143557</a></p>

<p>an optimization for filters returning a SEEK\_NEXT\_USING\_HINT would be to seek to the highest hint (Any previous/lower row won't be accepted by the filter returning that seek).</p>

<p>This JIRA is to explore this potential optimization.</p>

Description: <p>When running the small tests, we found TestHFileBackedByBucketCache failed randomly</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

mvn clean <span class="code-keyword">package</span> install -DrunSmallTests -Dtest=TestHFileBackedByBucketCache

Running org.apache.hadoop.hbase.io.hfile.TestHFileBackedByBucketCache

Tests run: 1, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 1.262 sec &lt;&lt;&lt; FAILURE! - in org.apache.hadoop.hbase.io.hfile.TestHFileBackedByBucketCache

testBucketCacheCachesAndPersists(org.apache.hadoop.hbase.io.hfile.TestHFileBackedByBucketCache) Time elapsed: 0.69 sec &lt;&lt;&lt; FAILURE!

java.lang.AssertionError: expected:&lt;5&gt; but was:&lt;4&gt;

at org.apache.hadoop.hbase.io.hfile.TestHFileBackedByBucketCache.testBucketCacheCachesAndPersists(TestHFileBackedByBucketCache.java:161)

</pre>

</div></div>

Description: <p>These days there are many tests timeout on builds.apache.org. I have no log on timeout tests but I find a possible reason: master crashes and MiniCluster will log "No master found; retry" forever until timeout.</p>

Description: <p>Otherwise we can not use it in secure environment.</p>

<p>Should be a netty handler, but see</p>

<p><a href="https://github.com/netty/netty/issues/1966" class="external-link" rel="nofollow">https://github.com/netty/netty/issues/1966</a></p>

<p>I do not think it will be available in the near future, so we need to do it by ourselves.</p>

Description: <p>This was what I did on cluster with 1.4.0-SNAPSHOT built Thursday:</p>

<p>Run 'hbase hbck -disableSplitAndMerge' on gateway node of the cluster<br/>

Terminate hbck early<br/>

Enter hbase shell where I observed:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

hbase(main):001:0&gt; splitormerge\_enabled 'SPLIT'

<span class="code-keyword">false</span>

0 row(s) in 0.3280 seconds

hbase(main):002:0&gt; splitormerge\_enabled 'MERGE'

<span class="code-keyword">false</span>

0 row(s) in 0.0070 seconds

</pre>

</div></div>

<p>Expectation is that the split / merge switches should be restored to default value after hbck exits.</p>

Description: <p>Corrects wrong default values for few options in the help message.</p>

<p>Final stats from multiple clients are intermingled making it hard to understand. Also the logged stats aren't very machine readable. It can be helpful in a daily perf testing rig which scraps logs for results.</p>

<p>Example of logs before the change.</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest latency log (microseconds), on 1048576 measures

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest latency log (microseconds), on 1048576 measures

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest latency log (microseconds), on 1048576 measures

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest latency log (microseconds), on 1048576 measures

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest latency log (microseconds), on 1048576 measures

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest latency log (microseconds), on 1048576 measures

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest latency log (microseconds), on 1048576 measures

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: 0/1048570/1048576, latency mean=953.98, min=359.00, max=324050.00, stdDev=851.82, 95th=1368.00, 99th=1625.00

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: 0/1048570/1048576, latency mean=953.92, min=356.00, max=323394.00, stdDev=817.55, 95th=1370.00, 99th=1618.00

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: 0/1048570/1048576, latency mean=953.98, min=367.00, max=322745.00, stdDev=840.43, 95th=1369.00, 99th=1622.00

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest latency log (microseconds), on 1048576 measures

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest latency log (microseconds), on 1048576 measures

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest latency log (microseconds), on 1048576 measures

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest Min = 375.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest Min = 363.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest Avg = 953.6624126434326

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest Avg = 953.4124526977539

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest StdDev = 781.3929776087633

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest StdDev = 742.8027916717297

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 50th = 894.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 50th = 894.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 75th = 1070.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 75th = 1071.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 95th = 1369.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 95th = 1369.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 99th = 1623.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 99th = 1624.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest Min = 372.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 99.9th = 3013.9980000000214

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest Avg = 953.2451229095459

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 99.9th = 3043.9980000000214

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest StdDev = 725.4744472152282

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 99.99th = 25282.380199996755

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 50th = 895.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 99.99th = 25812.763399994

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 75th = 1071.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 99.999th = 89772.78990004538

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 95th = 1369.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 99.999th = 122808.39587019826

</pre>

</div></div>

<p>After the change</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>16/03/06 02:38:21 INFO hbase.PerformanceEvaluation: Test : RandomWriteTest, Thread : TestClient-1

16/03/06 02:38:21 INFO hbase.PerformanceEvaluation: Latency (us) : , mean=6.23, min=2.00, max=101433.00, stdDev=246.62, 50th=2.00, 75th=2.00, 95th=3.00, 99th=13.00, 99.9th=558.00, 99.99th=9656.19, 99.999th=20213.63

16/03/06 02:38:21 INFO hbase.PerformanceEvaluation: Num measures (latency) : 1048576

16/03/06 02:38:21 INFO hbase.PerformanceEvaluation: ValueSize (bytes) : , mean=0.00, min=0.00, max=0.00, stdDev=0.00, 50th=0.00, 75th=0.00, 95th=0.00, 99th=0.00, 99.9th=0.00, 99.99th=0.00, 99.999th=0.00

16/03/06 02:38:21 INFO hbase.PerformanceEvaluation: Num measures (ValueSize): 0

...

16/03/06 02:38:21 INFO hbase.PerformanceEvaluation: Test : RandomWriteTest, Thread : TestClient-0

16/03/06 02:38:21 INFO hbase.PerformanceEvaluation: Latency (us) : , mean=6.23, min=1.00, max=99321.00, stdDev=246.41, 50th=2.00, 75th=2.00, 95th=3.00, 99th=13.00, 99.9th=566.00, 99.99th=9694.60, 99.999th=23311.86

....

</pre>

</div></div>

Description: <p>On running hbase pe --nomapred increment/append 10, i see the following output where it seems like threads are executing operations serially. In the UI too, only one RS is getting requests at a time.</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>6/03/05 22:26:15 INFO hbase.PerformanceEvaluation: Timed test starting in thread TestClient-1

16/03/05 22:26:15 INFO hbase.PerformanceEvaluation: Timed test starting in thread TestClient-2

16/03/05 22:26:15 INFO hbase.PerformanceEvaluation: Timed test starting in thread TestClient-8

16/03/05 22:26:15 INFO hbase.PerformanceEvaluation: Timed test starting in thread TestClient-0

16/03/05 22:26:15 INFO hbase.PerformanceEvaluation: Timed test starting in thread TestClient-4

16/03/05 22:26:15 INFO hbase.PerformanceEvaluation: Timed test starting in thread TestClient-6

16/03/05 22:26:15 INFO hbase.PerformanceEvaluation: Timed test starting in thread TestClient-7

16/03/05 22:26:15 INFO hbase.PerformanceEvaluation: Timed test starting in thread TestClient-5

16/03/05 22:26:15 INFO hbase.PerformanceEvaluation: Timed test starting in thread TestClient-9

16/03/05 22:26:15 INFO hbase.PerformanceEvaluation: Timed test starting in thread TestClient-3

16/03/05 22:27:54 INFO hbase.PerformanceEvaluation: 0/104857/1048576, latency mean=942.48, min=390.00, max=163444.00, stdDev=892.64, 95th=1361.00, 99th=1605.00

16/03/05 22:27:54 INFO hbase.PerformanceEvaluation: 0/104857/1048576, latency mean=942.53, min=366.00, max=163400.00, stdDev=885.49, 95th=1361.00, 99th=1605.00

16/03/05 22:27:54 INFO hbase.PerformanceEvaluation: 0/104857/1048576, latency mean=942.41, min=402.00, max=163436.00, stdDev=891.54, 95th=1359.00, 99th=1602.41

16/03/05 22:27:54 INFO hbase.PerformanceEvaluation: 0/104857/1048576, latency mean=942.51, min=399.00, max=163610.00, stdDev=892.40, 95th=1360.00, 99th=1600.00

16/03/05 22:27:54 INFO hbase.PerformanceEvaluation: 0/104857/1048576, latency mean=942.59, min=393.00, max=162932.00, stdDev=887.65, 95th=1361.00, 99th=1604.00

16/03/05 22:27:54 INFO hbase.PerformanceEvaluation: 0/104857/1048576, latency mean=942.26, min=385.00, max=163482.00, stdDev=891.71, 95th=1358.00, 99th=1599.00

16/03/05 22:27:54 INFO hbase.PerformanceEvaluation: 0/104857/1048576, latency mean=942.51, min=383.00, max=163246.00, stdDev=888.07, 95th=1360.00, 99th=1605.00

16/03/05 22:27:54 INFO hbase.PerformanceEvaluation: 0/104857/1048576, latency mean=942.45, min=385.00, max=163405.00, stdDev=886.65, 95th=1359.00, 99th=1604.00

16/03/05 22:27:54 INFO hbase.PerformanceEvaluation: 0/104857/1048576, latency mean=942.38, min=400.00, max=163580.00, stdDev=887.28, 95th=1359.00, 99th=1602.00

16/03/05 22:27:54 INFO hbase.PerformanceEvaluation: 0/104857/1048576, latency mean=942.29, min=407.00, max=163403.00, stdDev=889.77, 95th=1357.00, 99th=1597.00

16/03/05 22:29:35 INFO hbase.PerformanceEvaluation: 0/209714/1048576, latency mean=950.75, min=366.00, max=163400.00, stdDev=817.84, 95th=1363.00, 99th=1605.00

16/03/05 22:29:35 INFO hbase.PerformanceEvaluation: 0/209714/1048576, latency mean=950.75, min=383.00, max=163246.00, stdDev=821.95, 95th=1363.00, 99th=1604.00

16/03/05 22:29:35 INFO hbase.PerformanceEvaluation: 0/209714/1048576, latency mean=950.75, min=389.00, max=163444.00, stdDev=824.03, 95th=1364.00, 99th=1603.00

16/03/05 22:29:35 INFO hbase.PerformanceEvaluation: 0/209714/1048576, latency mean=950.56, min=382.00, max=163403.00, stdDev=822.44, 95th=1363.00, 99th=1603.00

16/03/05 22:29:35 INFO hbase.PerformanceEvaluation: 0/209714/1048576, latency mean=950.79, min=393.00, max=162932.00, stdDev=818.75, 95th=1365.00, 99th=1601.84

16/03/05 22:29:35 INFO hbase.PerformanceEvaluation: 0/209714/1048576, latency mean=950.70, min=388.00, max=163436.00, stdDev=823.52, 95th=1364.00, 99th=1606.00

16/03/05 22:29:35 INFO hbase.PerformanceEvaluation: 0/209714/1048576, latency mean=950.72, min=376.00, max=163405.00, stdDev=820.65, 95th=1364.00, 99th=1605.00

16/03/05 22:29:35 INFO hbase.PerformanceEvaluation: 0/209714/1048576, latency mean=950.56, min=382.00, max=163482.00, stdDev=823.43, 95th=1363.00, 99th=1599.00

16/03/05 22:29:35 INFO hbase.PerformanceEvaluation: 0/209714/1048576, latency mean=950.67, min=376.00, max=163580.00, stdDev=821.59, 95th=1364.00, 99th=1602.00

16/03/05 22:29:35 INFO hbase.PerformanceEvaluation: 0/209714/1048576, latency mean=950.77, min=390.00, max=163610.00, stdDev=823.88, 95th=1363.00, 99th=1600.00

16/03/05 22:31:16 INFO hbase.PerformanceEvaluation: 0/314571/1048576, latency mean=952.21, min=369.00, max=162932.00, stdDev=787.36, 95th=1361.00, 99th=1595.00

16/03/05 22:31:16 INFO hbase.PerformanceEvaluation: 0/314571/1048576, latency mean=952.18, min=374.00, max=163610.00, stdDev=791.11, 95th=1359.00, 99th=1594.00

16/03/05 22:31:16 INFO hbase.PerformanceEvaluation: 0/314571/1048576, latency mean=952.30, min=367.00, max=163444.00, stdDev=802.21, 95th=1362.00, 99th=1597.00

16/03/05 22:31:16 INFO hbase.PerformanceEvaluation: 0/314571/1048576, latency mean=952.38, min=366.00, max=163400.00, stdDev=799.61, 95th=1360.00, 99th=1596.00

16/03/05 22:31:16 INFO hbase.PerformanceEvaluation: 0/314571/1048576, latency mean=952.31, min=375.00, max=163580.00, stdDev=802.77, 95th=1359.00, 99th=1596.00

16/03/05 22:31:16 INFO hbase.PerformanceEvaluation: 0/314571/1048576, latency mean=952.12, min=388.00, max=163436.00, stdDev=791.34, 95th=1361.00, 99th=1598.00

16/03/05 22:31:16 INFO hbase.PerformanceEvaluation: 0/314571/1048576, latency mean=952.13, min=368.00, max=163405.00, stdDev=788.29, 95th=1360.00, 99th=1598.00

16/03/05 22:31:16 INFO hbase.PerformanceEvaluation: 0/314571/1048576, latency mean=952.35, min=383.00, max=163246.00, stdDev=801.26, 95th=1362.00, 99th=1599.00

16/03/05 22:31:16 INFO hbase.PerformanceEvaluation: 0/314571/1048576, latency mean=952.21, min=382.00, max=163403.00, stdDev=801.88, 95th=1359.00, 99th=1598.00

</pre>

</div></div>

Description: <p>hbase pe --nomapred --rows=100 --table='t4' randomWrite 10</p>

<ol>

<li>count on t4 gives 620 rows</li>

</ol>

<p>hbase pe --nomapred --rows=200 --table='t5' randomWrite 10</p>

<ol>

<li>count on t5 gives 1257 rows</li>

</ol>

<p>hbase pe --nomapred --table='t6' --rows=200 randomWrite 1</p>

<ol>

<li>count on t6 gives 126 rows</li>

</ol>

<p>I was working with 1.2.0, but it's likely that it'll also be affecting master.</p>

Description:

Description:

Description: <p>When we compact, we can output multiple files along the current window boundaries. There are two use cases:</p>

<p>1. Major compaction: We want to output date tiered store files with data older than max age archived in trunks of the window size on the higher tier.<br/>

2. Bulk load files and the old file generated by major compaction before upgrading to DTCP.</p>

<p>Pros: <br/>

1. Restore locality, process versioning, updates and deletes while maintaining the tiered layout.<br/>

2. The best way to fix a skewed layout.</p>

<p>This work is based on a prototype of DateTieredCompactor from <a href="https://issues.apache.org/jira/browse/HBASE-15389" title="Write out multiple files when compaction" class="issue-link" data-issue-key="HBASE-15389">HBASE-15389</a> and focused on the part to meet needs for these two use cases while supporting others. I have to call out a few design decisions:</p>

<p>1. We only want to output the files along all windows for major compaction. And we want to output multiple files older than max age in the trunks of the maximum tier window size determined by base window size, windows per tier and max age.</p>

<p>2. For minor compaction, we don't want to output too many files, which will remain around because of current restriction of contiguous compaction by seq id. I will only output two files if all the files in the windows are being combined, one for the data within window and the other for the out-of-window tail. If there is any file in the window excluded from compaction, only one file will be output from compaction. When the windows are promoted, the situation of out of order data will gradually improve. For the incoming window, we need to accommodate the case with user-specified future data.</p>

<p>3. We have to pass the boundaries with the list of store file as a complete time snapshot instead of two separate calls because window layout is determined by the time the computation is called. So we will need new type of compaction request. </p>

<p>4. Since we will assign the same seq id for all output files, we need to sort by maxTimestamp subsequently. Right now all compaction policy gets the files sorted for StoreFileManager which sorts by seq id and other criteria. I will use this order for DTCP only, to avoid impacting other compaction policies. </p>

<p>5. We need some cleanup of current design of StoreEngine and CompactionPolicy.</p>

Description: <p>Sound like bulkLoad methods are there only on the Scala side. Doesn't seems to be doable on the Java side. Need to add it into JavaHBaseContext.</p>

Description: <p>In RegionScannerImpl, we have two heaps, storeHeap and joinedHeap. If we have a filter and it doesn't apply to all cf, the stores whose families needn't be filtered will be in joinedHeap. We scan storeHeap first, then joinedHeap, and merge the results and sort and return to client. We need sort because the order of Cell is rowkey/cf/cq/ts and a smaller cf may be in the joinedHeap.</p>

<p>However, after <a href="https://issues.apache.org/jira/browse/HBASE-11544" title="[Ergonomics] hbase.client.scanner.caching is dogged and will try to return batch even if it means OOME" class="issue-link" data-issue-key="HBASE-11544"><del>HBASE-11544</del></a> we may transfer partial results when we get SIZE\_LIMIT\_REACHED\_MID\_ROW or other similar states. We may return a larger cf first because it is in storeHeap and then a smaller cf because it is in joinedHeap. Server won't hold all cells in a row and client doesn't have a sorting logic. The order of cf in Result for user is wrong.</p>

<p>And a more critical bug is, if we get a LIMIT\_REACHED\_MID\_ROW on the last cell of a row in storeHeap, we will break scanning in RegionScannerImpl and in populateResult we will change the state to SIZE\_LIMIT\_REACHED because next peeked cell is next row. But this is only the last cell of one and we have two... And SIZE\_LIMIT\_REACHED means this Result is not partial (by ScannerContext.partialResultFormed), client will see it and merge them and return to user with losing data of joinedHeap. On next scan we will read next row of storeHeap and joinedHeap is forgotten and never be read...</p>

Description: <p>Create bulk load replication znode(hfile-refs) in ZK replication queue by default same as hbase replication znode. <br/>

Otherwise the problem what happens is currently replication admin directly operates on ZK without routing through HM/RS. So suppose if a user enables the replication for bulk loaded data in server but fails to do the same in the client configurations then add peer will not add hfile-refs znode, resulting in replication failure for bulk loaded data.<br/>

So after fixing this the behavior will be same as mutation replication.</p>

Description: <p>When troubleshooting HBase-input MR job map failures, the TableSplit's toString() definition comes very handy. However, it still does not tell us directly which region ID it was created for (it does tell us the key and the host, but not the exact region name, so there's a second lookup needed from the UI to find the ID to then check the servers with).</p>

<p>It would be beneficial to have the encoded region name as part of the output, so this extra manual lookup can be avoided. The encoded name is much easier to find events with.</p>

Description: <p>tried reloading the page and clicking on the tables, shows nothing, though regions are assigned/</p>

Description: <p>This is the case where daughter region A is created successfully under table dir but region server crashed during creating daughter region B. <br/>

So split failed and left an orphan daughter dir in the filesystem.</p>

<p>Currently we are just clearing the ".split" directory not the orphan daughter regions which might be moved out under the table directory during previous failed SPLIT operation.</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

void cleanupAnySplitDetritus() <span class="code-keyword">throws</span> IOException {

Path splitdir = <span class="code-keyword">this</span>.getSplitsDir();

<span class="code-keyword">if</span> (!fs.exists(splitdir)) <span class="code-keyword">return</span>;

<span class="code-comment">// Look at the splitdir. It could have the encoded names of the daughter

</span> <span class="code-comment">// regions we tried to make. See <span class="code-keyword">if</span> the daughter regions actually got made

</span> <span class="code-comment">// out under the tabledir. If here under splitdir still, then the split did

</span> <span class="code-comment">// not complete. Try and <span class="code-keyword">do</span> cleanup. This code WILL NOT <span class="code-keyword">catch</span> the <span class="code-keyword">case</span>

</span> <span class="code-comment">// where we successfully created daughter a but regionserver crashed during

</span> <span class="code-comment">// the creation of region b. In <span class="code-keyword">this</span> <span class="code-keyword">case</span>, there'll be an orphan daughter

</span> <span class="code-comment">// dir in the filesystem. TOOD: Fix.

</span> FileStatus[] daughters = FSUtils.listStatus(fs, splitdir, <span class="code-keyword">new</span> FSUtils.DirFilter(fs));

<span class="code-keyword">if</span> (daughters != <span class="code-keyword">null</span>) {

<span class="code-keyword">for</span> (FileStatus daughter: daughters) {

Path daughterDir = <span class="code-keyword">new</span> Path(getTableDir(), daughter.getPath().getName());

<span class="code-keyword">if</span> (fs.exists(daughterDir) &amp;&amp; !deleteDir(daughterDir)) {

<span class="code-keyword">throw</span> <span class="code-keyword">new</span> IOException(<span class="code-quote">"Failed delete of "</span> + daughterDir);

}

}

}

cleanupSplitsDir();

LOG.info(<span class="code-quote">"Cleaned up old failed split transaction detritus: "</span> + splitdir);

}

</pre>

</div></div>

<p>Creating this JIRA is to track these changes.</p>

Description: <p>Enable table replication command will fail when parent znode is not /hbase(default) in peer cluster and there is only one peer cluster added in the source cluster.</p>

Description: <p>As found by Daniel "SystemTap" Pol, a simple Get results in our reading two HFileBlocks, the one that contains the wanted Cell, and the block that follows.</p>

<p>Here is a bit of custom logging that logs a stack trace on each HFileBlock read so you can see the call stack responsible:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

2016-03-03 22:20:30,191 INFO [B.defaultRpcServer.handler=20,queue=2,port=16020] regionserver.StoreScanner: START LOOP

2016-03-03 22:20:30,192 INFO [B.defaultRpcServer.handler=20,queue=2,port=16020] regionserver.StoreScanner: QCODE SEEK\_NEXT\_COL

2016-03-03 22:20:30,192 INFO [B.defaultRpcServer.handler=20,queue=2,port=16020] hfile.HFileBlockIndex: STARTED WHILE

2016-03-03 22:20:30,192 INFO [B.defaultRpcServer.handler=20,queue=2,port=16020] hfile.CombinedBlockCache: OUT OF L2

2016-03-03 22:20:30,192 TRACE [B.defaultRpcServer.handler=20,queue=2,port=16020] bucket.BucketCache: Read offset=31409152, len=2103

2016-03-03 22:20:30,192 TRACE [B.defaultRpcServer.handler=20,queue=2,port=16020] bucket.FileIOEngine: offset=31409152, length=2103

2016-03-03 22:20:30,193 TRACE [B.defaultRpcServer.handler=20,queue=2,port=16020] hfile.HFileReaderImpl: From Cache [blockType=DATA, fileOffset=2055421, headerSize=33, onDiskSizeWithoutHeader=2024, uncompressedSizeWithoutHeader=2020, prevBlockOffset=2053364, isUseHBaseChecksum=<span class="code-keyword">true</span>, checksumType=CRC32C, bytesPerChecksum=16384, onDiskDataSizeWithHeader=2053, getOnDiskSizeWithHeader=2057, totalChecksumBytes=4, isUnpacked=<span class="code-keyword">true</span>, buf=[org.apache.hadoop.hbase.nio.SingleByteBuff@e19fbd54], dataBeginsWith=\x00\x00\x00)\x00\x00\x01`\x00\x16user995139035672819231, fileContext=[usesHBaseChecksum=<span class="code-keyword">true</span>, checksumType=CRC32C, bytesPerChecksum=16384, blocksize=65536, encoding=NONE, includesMvcc=<span class="code-keyword">true</span>, includesTags=<span class="code-keyword">false</span>, compressAlgo=NONE, compressTags=<span class="code-keyword">false</span>, cryptoContext=[cipher=NONE keyHash=NONE]]]

2016-03-03 22:20:30,193 TRACE [B.defaultRpcServer.handler=20,queue=2,port=16020] hfile.HFileReaderImpl: Cache hit <span class="code-keyword">return</span> [blockType=DATA, fileOffset=2055421, headerSize=33, onDiskSizeWithoutHeader=2024, uncompressedSizeWithoutHeader=2020, prevBlockOffset=2053364, isUseHBaseChecksum=<span class="code-keyword">true</span>, checksumType=CRC32C, bytesPerChecksum=16384, onDiskDataSizeWithHeader=2053, getOnDiskSizeWithHeader=2057, totalChecksumBytes=4, isUnpacked=<span class="code-keyword">true</span>, buf=[org.apache.hadoop.hbase.nio.SingleByteBuff@e19fbd54], dataBeginsWith=\x00\x00\x00)\x00\x00\x01`\x00\x16user995139035672819231, fileContext=[usesHBaseChecksum=<span class="code-keyword">true</span>, checksumType=CRC32C, bytesPerChecksum=16384, blocksize=65536, encoding=NONE, includesMvcc=<span class="code-keyword">true</span>, includesTags=<span class="code-keyword">false</span>, compressAlgo=NONE, compressTags=<span class="code-keyword">false</span>, cryptoContext=[cipher=NONE keyHash=NONE]]]

java.lang.Throwable

at org.apache.hadoop.hbase.io.hfile.HFileReaderImpl.readBlock(HFileReaderImpl.java:1515)

at org.apache.hadoop.hbase.io.hfile.HFileBlockIndex$CellBasedKeyBlockIndexReader.loadDataBlockWithScanInfo(HFileBlockIndex.java:324)

at org.apache.hadoop.hbase.io.hfile.HFileReaderImpl$HFileScannerImpl.seekTo(HFileReaderImpl.java:831)

at org.apache.hadoop.hbase.io.hfile.HFileReaderImpl$HFileScannerImpl.reseekTo(HFileReaderImpl.java:812)

at org.apache.hadoop.hbase.regionserver.StoreFileScanner.reseekAtOrAfter(StoreFileScanner.java:288)

at org.apache.hadoop.hbase.regionserver.StoreFileScanner.reseek(StoreFileScanner.java:198)

at org.apache.hadoop.hbase.regionserver.NonLazyKeyValueScanner.doRealSeek(NonLazyKeyValueScanner.java:54)

at org.apache.hadoop.hbase.regionserver.KeyValueHeap.generalizedSeek(KeyValueHeap.java:321)

at org.apache.hadoop.hbase.regionserver.KeyValueHeap.requestSeek(KeyValueHeap.java:279)

at org.apache.hadoop.hbase.regionserver.StoreScanner.reseek(StoreScanner.java:806)

at org.apache.hadoop.hbase.regionserver.StoreScanner.seekAsDirection(StoreScanner.java:795)

at org.apache.hadoop.hbase.regionserver.StoreScanner.next(StoreScanner.java:624)

at org.apache.hadoop.hbase.regionserver.KeyValueHeap.next(KeyValueHeap.java:153)

at org.apache.hadoop.hbase.regionserver.HRegion$RegionScannerImpl.populateResult(HRegion.java:5703)

at org.apache.hadoop.hbase.regionserver.HRegion$RegionScannerImpl.nextInternal(HRegion.java:5849)

at org.apache.hadoop.hbase.regionserver.HRegion$RegionScannerImpl.nextRaw(HRegion.java:5622)

at org.apache.hadoop.hbase.regionserver.HRegion$RegionScannerImpl.next(HRegion.java:5598)

at org.apache.hadoop.hbase.regionserver.HRegion$RegionScannerImpl.next(HRegion.java:5584)

at org.apache.hadoop.hbase.regionserver.RSRpcServices.get(RSRpcServices.java:2187)

at org.apache.hadoop.hbase.regionserver.RSRpcServices.get(RSRpcServices.java:2123)

at org.apache.hadoop.hbase.protobuf.generated.ClientProtos$ClientService$2.callBlockingMethod(ClientProtos.java:33512)

at org.apache.hadoop.hbase.ipc.RpcServer.call(RpcServer.java:2229)

at org.apache.hadoop.hbase.ipc.CallRunner.run(CallRunner.java:109)

at org.apache.hadoop.hbase.ipc.RpcExecutor.consumerLoop(RpcExecutor.java:136)

at org.apache.hadoop.hbase.ipc.RpcExecutor$1.run(RpcExecutor.java:111)

at java.lang.<span class="code-object">Thread</span>.run(<span class="code-object">Thread</span>.java:745)

2016-03-03 22:20:30,193 INFO [B.defaultRpcServer.handler=20,queue=2,port=16020] hfile.HFileBlockIndex: READ [blockType=DATA, fileOffset=2055421, headerSize=33, onDiskSizeWithoutHeader=2024, uncompressedSizeWithoutHeader=2020, prevBlockOffset=2053364, isUseHBaseChecksum=<span class="code-keyword">true</span>, checksumType=CRC32C, bytesPerChecksum=16384, onDiskDataSizeWithHeader=2053, getOnDiskSizeWithHeader=2057, totalChecksumBytes=4, isUnpacked=<span class="code-keyword">true</span>, buf=[org.apache.hadoop.hbase.nio.SingleByteBuff@e19fbd54], dataBeginsWith=\x00\x00\x00)\x00\x00\x01`\x00\x16user995139035672819231, fileContext=[usesHBaseChecksum=<span class="code-keyword">true</span>, checksumType=CRC32C, bytesPerChecksum=16384, blocksize=65536, encoding=NONE, includesMvcc=<span class="code-keyword">true</span>, includesTags=<span class="code-keyword">false</span>, compressAlgo=NONE, compressTags=<span class="code-keyword">false</span>, cryptoContext=[cipher=NONE keyHash=NONE]]]

2016-03-03 22:20:30,193 INFO [B.defaultRpcServer.handler=20,queue=2,port=16020] hfile.HFileBlockIndex: DATA BLOCK IS TRUE BREAK

2016-03-03 22:20:30,193 INFO [B.defaultRpcServer.handler=20,queue=2,port=16020] hfile.HFileBlockIndex: RETURNING

2016-03-03 22:20:30,194 INFO [B.defaultRpcServer.handler=20,queue=2,port=16020] regionserver.StoreScanner: QCODE INCLUDE

2016-03-03 22:20:30,194 INFO [B.defaultRpcServer.handler=20,queue=2,port=16020] regionserver.StoreScanner: QCODE SKIP

2016-03-03 22:20:30,194 INFO [B.defaultRpcServer.handler=20,queue=2,port=16020] hfile.CombinedBlockCache: OUT OF L2

2016-03-03 22:20:30,194 TRACE [B.defaultRpcServer.handler=20,queue=2,port=16020] bucket.BucketCache: Read offset=30932992, len=2103

2016-03-03 22:20:30,194 TRACE [B.defaultRpcServer.handler=20,queue=2,port=16020] bucket.FileIOEngine: offset=30932992, length=2103

2016-03-03 22:20:30,194 TRACE [B.defaultRpcServer.handler=20,queue=2,port=16020] hfile.HFileReaderImpl: From Cache [blockType=DATA, fileOffset=2057478, headerSize=33, onDiskSizeWithoutHeader=2024, uncompressedSizeWithoutHeader=2020, prevBlockOffset=2055421, isUseHBaseChecksum=<span class="code-keyword">true</span>, checksumType=CRC32C, bytesPerChecksum=16384, onDiskDataSizeWithHeader=2053, getOnDiskSizeWithHeader=2057, totalChecksumBytes=4, isUnpacked=<span class="code-keyword">true</span>, buf=[org.apache.hadoop.hbase.nio.SingleByteBuff@6063ac95], dataBeginsWith=\x00\x00\x00)\x00\x00\x01`\x00\x16user995698996184959679, fileContext=[usesHBaseChecksum=<span class="code-keyword">true</span>, checksumType=CRC32C, bytesPerChecksum=16384, blocksize=65536, encoding=NONE, includesMvcc=<span class="code-keyword">true</span>, includesTags=<span class="code-keyword">false</span>, compressAlgo=NONE, compressTags=<span class="code-keyword">false</span>, cryptoContext=[cipher=NONE keyHash=NONE]]]

2016-03-03 22:20:30,194 TRACE [B.defaultRpcServer.handler=20,queue=2,port=16020] hfile.HFileReaderImpl: Cache hit <span class="code-keyword">return</span> [blockType=DATA, fileOffset=2057478, headerSize=33, onDiskSizeWithoutHeader=2024, uncompressedSizeWithoutHeader=2020, prevBlockOffset=2055421, isUseHBaseChecksum=<span class="code-keyword">true</span>, checksumType=CRC32C, bytesPerChecksum=16384, onDiskDataSizeWithHeader=2053, getOnDiskSizeWithHeader=2057, totalChecksumBytes=4, isUnpacked=<span class="code-keyword">true</span>, buf=[org.apache.hadoop.hbase.nio.SingleByteBuff@6063ac95], dataBeginsWith=\x00\x00\x00)\x00\x00\x01`\x00\x16user995698996184959679, fileContext=[usesHBaseChecksum=<span class="code-keyword">true</span>, checksumType=CRC32C, bytesPerChecksum=16384, blocksize=65536, encoding=NONE, includesMvcc=<span class="code-keyword">true</span>, includesTags=<span class="code-keyword">false</span>, compressAlgo=NONE, compressTags=<span class="code-keyword">false</span>, cryptoContext=[cipher=NONE keyHash=NONE]]]

java.lang.Throwable

at org.apache.hadoop.hbase.io.hfile.HFileReaderImpl.readBlock(HFileReaderImpl.java:1515)

at org.apache.hadoop.hbase.io.hfile.HFileReaderImpl$HFileScannerImpl.readNextDataBlock(HFileReaderImpl.java:906)

at org.apache.hadoop.hbase.io.hfile.HFileReaderImpl$HFileScannerImpl.isNextBlock(HFileReaderImpl.java:1106)

at org.apache.hadoop.hbase.io.hfile.HFileReaderImpl$HFileScannerImpl.positionForNextBlock(HFileReaderImpl.java:1100)

at org.apache.hadoop.hbase.io.hfile.HFileReaderImpl$HFileScannerImpl.\_next(HFileReaderImpl.java:1118)

at org.apache.hadoop.hbase.io.hfile.HFileReaderImpl$HFileScannerImpl.next(HFileReaderImpl.java:1139)

at org.apache.hadoop.hbase.regionserver.StoreFileScanner.next(StoreFileScanner.java:152)

at org.apache.hadoop.hbase.regionserver.KeyValueHeap.next(KeyValueHeap.java:114)

at org.apache.hadoop.hbase.regionserver.StoreScanner.next(StoreScanner.java:628)

at org.apache.hadoop.hbase.regionserver.KeyValueHeap.next(KeyValueHeap.java:153)

at org.apache.hadoop.hbase.regionserver.HRegion$RegionScannerImpl.populateResult(HRegion.java:5703)

at org.apache.hadoop.hbase.regionserver.HRegion$RegionScannerImpl.nextInternal(HRegion.java:5849)

at org.apache.hadoop.hbase.regionserver.HRegion$RegionScannerImpl.nextRaw(HRegion.java:5622)

at org.apache.hadoop.hbase.regionserver.HRegion$RegionScannerImpl.next(HRegion.java:5598)

at org.apache.hadoop.hbase.regionserver.HRegion$RegionScannerImpl.next(HRegion.java:5584)

at org.apache.hadoop.hbase.regionserver.RSRpcServices.get(RSRpcServices.java:2187)

at org.apache.hadoop.hbase.regionserver.RSRpcServices.get(RSRpcServices.java:2123)

at org.apache.hadoop.hbase.protobuf.generated.ClientProtos$ClientService$2.callBlockingMethod(ClientProtos.java:33512)

at org.apache.hadoop.hbase.ipc.RpcServer.call(RpcServer.java:2229)

at org.apache.hadoop.hbase.ipc.CallRunner.run(CallRunner.java:109)

at org.apache.hadoop.hbase.ipc.RpcExecutor.consumerLoop(RpcExecutor.java:136)

at org.apache.hadoop.hbase.ipc.RpcExecutor$1.run(RpcExecutor.java:111)

at java.lang.<span class="code-object">Thread</span>.run(<span class="code-object">Thread</span>.java:745)

2016-03-03 22:20:30,195 INFO [B.defaultRpcServer.handler=20,queue=2,port=16020] regionserver.StoreScanner: QCODE DONE

</pre>

</div></div>

<p>We are in StoreScanner#next.</p>

<p>Matcher does SEEK\_NEXT\_COL.<br/>

We pull in the block that has our Cell in it.<br/>

Matcher does INCLUDE<br/>

Then Matcher does SKIP<br/>

SKIP has us go read the next block.<br/>

Then Matcher does DONE.</p>

<p>Why ain't I getting to DONE after INCLUDE?</p>

<p>Any clues?</p>

<p>This code has been like this a while.</p>

Description: <p>When deleting a large table in HBase, there will be a large info log in HMaster.</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

2016-02-29,05:58:45,920 INFO org.apache.hadoop.hbase.catalog.MetaEditor: Deleted [{ENCODED =&gt; 4b54572150941cd03f5addfdeab0a754, NAME =&gt; 'YCSBTest,,1453186492932.4b54572150941cd03f5addfdeab0a754.', STARTKEY =&gt; '', ENDKEY =&gt; 'user01'}, {ENCODED =&gt; 715e142bcd6a31d7842abf286ef8a5fe, NAME =&gt; 'YCSBTest,user01,1453186492933.715e142bcd6a31d7842abf286ef8a5fe.', STARTKEY =&gt; 'user01', ENDKEY =&gt; 'user02'}, {ENCODED =&gt; 5f9cef5714973f13baa63fba29a68d70, NAME =&gt; 'YCSBTest,user02,1453186492933.5f9cef5714973f13baa63fba29a68d70.', STARTKEY =&gt; 'user02', ENDKEY =&gt; 'user03'}, {ENCODED =&gt; 86cf3fa4c0a6b911275512c1d4b78533, NAME =&gt; 'YCSBTest,user0...

</pre>

</div></div>

<p>The reason is that MetaTableAccessor will log all regions when deleting them from meta. See, MetaTableAccessor.java#deleteRegions</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

<span class="code-keyword">public</span> <span class="code-keyword">static</span> void deleteRegions(Connection connection,

List&lt;HRegionInfo&gt; regionsInfo, <span class="code-object">long</span> ts) <span class="code-keyword">throws</span> IOException {

List&lt;Delete&gt; deletes = <span class="code-keyword">new</span> ArrayList&lt;Delete&gt;(regionsInfo.size());

<span class="code-keyword">for</span> (HRegionInfo hri: regionsInfo) {

Delete e = <span class="code-keyword">new</span> Delete(hri.getRegionName());

e.addFamily(getCatalogFamily(), ts);

deletes.add(e);

}

deleteFromMetaTable(connection, deletes);

LOG.info(<span class="code-quote">"Deleted "</span> + regionsInfo);

}

</pre>

</div></div>

<p>Just change the info log to debug and add a info log about the number of deleted regions. Others suggestions are welcomed~</p>

Description: <p>In #isMetaClearingException() we're checking for CallQueueTooBigException, but under debugger in tests I saw that what we're really getting in here is RemoteWithExtrasException, which doesn't allow to easily unwrap CQTBE from it, since it's stored in the classname field, and findException() or unwrapRemoteException() fail to unwrap it correctly.</p>

<p>I think we'd have the same behavior with other exceptions wrapper this way.</p>

Description:

Description: <p>Some new methods and some old ones do not have ACLs. </p>

<p>A basic look at the master rpc endpoints results in </p>

<ul class="alternate" type="square">

<li>Catalog janitor methods</li>

<li>set balancer switch</li>

<li>Normalizer methods</li>

<li>split merge switch</li>

<li>mob methods</li>

<li>others?</li>

</ul>

Description: <p>Currently we are initializing HalfStoreFileReader to split the HFile but we might have different implementation for splitting. So we can make it configurable. It's needed for local indexing in Phoenix(<a href="https://issues.apache.org/jira/browse/PHOENIX-2736" title="Fix possible data loss with local indexes when there are splits during bulkload" class="issue-link" data-issue-key="PHOENIX-2736">PHOENIX-2736</a>). </p>

Description: <p>We use the global flag hbase.rs.prefetchblocksonopen only and ignore the HCD setting.</p>

<p>Purge from HCD or hook it up again (it probably worked once).</p>

<p>Thanks to Daniel Pol for finding this one. Let me fix.</p>

Description: <p>When using Wsab file system, we found that a failed atomic folder rename operation can never recovery for the destination file deleted in Wasb filesystem. </p>

{quota}

<p>ls: Attempting to complete rename of file hbase/azurtst-xiaomi/data/default/YCSBTest/.tabledesc during folder rename redo, and file was not found in source or destination.</p>

<blockquote></blockquote>

<p>The reason is the the file is renamed to the destination file before the crash, and the destination file is deleted by another process after crash. So the recovery is blocked during finishing the rename operation of this file when found the source and destination files all don't exist.</p>

<p>See: NativeAzureFileSystem.java #finishSingleFileRename</p>

<p>Another serious problem is that the recovery of atomic rename operation may delete new created file which is same name as the source file, because the file system don't check if there are rename operation need be redo.</p>

<p>Suggestions are welcomed~</p>

Description:

Description: <p>Right now, we always hit the primary replica for meta and fallback to the secondary replicas in case of a timeout. This can hamper performance in scenarios where meta becomes a hot region e.g. cluster ramp up..clients dropping connections etc.</p>

<p>It's good to have a load distribution approach on meta's secondary replicas with fallback to primary if we read stale data.</p>

Description: <p>This feature was removed from the base patch. So we can come up with a proper interface for other components to use as well, as directly accessing jmx is not an option.</p>

Description: <p>In MOB, there is a periodical compaction which runs in HMaster (It can be disabled by configuration), some small mob files are merged into bigger ones. Now the compaction only runs in HMaster which is not efficient and might impact the running of HMaster. In this JIRA, a distributed MOB compaction is introduced, it is triggered by HMaster, but all the compaction jobs are distributed to HRegionServers.</p>

Description: <p>Rollback is no longer needed after "<a href="https://issues.apache.org/jira/browse/HBASE-15158" title="Change order in which we do write pipeline operations; do all under row locks!" class="issue-link" data-issue-key="HBASE-15158"><del>HBASE-15158</del></a> Change order in which we do write pipeline operations; do all under row locks". Purge support. Will simplify this segment work.</p>

Description: <p>This issue found by <a href="https://issues.apache.org/jira/secure/ViewProfile.jspa?name=appy" class="user-hover" rel="appy">Appy</a>. In <a href="https://issues.apache.org/jira/browse/HBASE-14099" title="StoreFile.passesKeyRangeFilter need not create Cells from the Scan&#39;s start and stop Row" class="issue-link" data-issue-key="HBASE-14099"><del>HBASE-14099</del></a> he says,<br/>

I was doing some testing when I hit a weird issue, seems related to this, so re-opening it (apologies in advance if it's not). Here's the stack trace</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>java.io.IOException: java.lang.UnsupportedOperationException: Cell is not of type org.apache.hadoop.hbase.SettableSequenceId

at org.apache.hadoop.hbase.CellUtil.setSequenceId(CellUtil.java:923)

at org.apache.hadoop.hbase.regionserver.StoreFileScanner.setCurrentCell(StoreFileScanner.java:231)

at org.apache.hadoop.hbase.regionserver.StoreFileScanner.requestSeek(StoreFileScanner.java:389)

at org.apache.hadoop.hbase.regionserver.StoreScanner.seekScanners(StoreScanner.java:348)

at org.apache.hadoop.hbase.regionserver.StoreScanner.&lt;init&gt;(StoreScanner.java:212)

at org.apache.hadoop.hbase.regionserver.HStore.createScanner(HStore.java:1873)

at org.apache.hadoop.hbase.regionserver.HStore.getScanner(HStore.java:1863)

at org.apache.hadoop.hbase.regionserver.HRegion$RegionScannerImpl.&lt;init&gt;(HRegion.java:5487)

at org.apache.hadoop.hbase.regionserver.HRegion.instantiateRegionScanner(HRegion.java:2577)

at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:2563)

at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:2544)

at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:2534)

at org.apache.hadoop.hbase.regionserver.HRegion.get(HRegion.java:6659)

at org.apache.hadoop.hbase.regionserver.HRegion.get(HRegion.java:6624)

at org.apache.hadoop.hbase.regionserver.TestWithSingleHRegion.test(TestWithSingleHRegion.java:48)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:606)

at org.junit.runners.model.FrameworkMethod$1.runReflectiveCall(FrameworkMethod.java:50)

at org.junit.internal.runners.model.ReflectiveCallable.run(ReflectiveCallable.java:12)

at org.junit.runners.model.FrameworkMethod.invokeExplosively(FrameworkMethod.java:47)

at org.junit.internal.runners.statements.InvokeMethod.evaluate(InvokeMethod.java:17)

at org.junit.runners.ParentRunner.runLeaf(ParentRunner.java:325)

at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:78)

at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:57)

at org.junit.runners.ParentRunner$3.run(ParentRunner.java:290)

at org.junit.runners.ParentRunner$1.schedule(ParentRunner.java:71)

at org.junit.runners.ParentRunner.runChildren(ParentRunner.java:288)

at org.junit.runners.ParentRunner.access$000(ParentRunner.java:58)

at org.junit.runners.ParentRunner$2.evaluate(ParentRunner.java:268)

at org.junit.runners.ParentRunner.run(ParentRunner.java:363)

at org.junit.runner.JUnitCore.run(JUnitCore.java:137)

at com.intellij.junit4.JUnit4IdeaTestRunner.startRunnerWithArgs(JUnit4IdeaTestRunner.java:117)

at com.intellij.rt.execution.junit.JUnitStarter.prepareStreamsAndStart(JUnitStarter.java:234)

at com.intellij.rt.execution.junit.JUnitStarter.main(JUnitStarter.java:74)

</pre>

</div></div>

<p>I think it's because of using changing from KeyValue to a different sub-class of <tt>Cell}}l which doesn't implement {{SettableSequenceId</tt></p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>- this.startKey = KeyValueUtil.createFirstDeleteFamilyOnRow(scan.getStartRow(),

+ this.startKey = CellUtil.createFirstDeleteFamilyCellOnRow(scan.getStartRow(),

</pre>

</div></div>

<p>To replicate it, download the attached hfiles somewhere, copy the TestWithSingleHRegion class to regionserver tests, change the ROOT\_DIR appropriately and run it.</p>

Description: <p>When a RS scanner get a TIME\_LIMIT\_REACHED\_MID\_ROW state, they will stop scanning and send back what it has read to client and mark the message as a heartbeat message. If there is no cell has been read, it will be an empty response. </p>

<p>However, ClientScanner only handles the situation that the client gets an empty heartbeat and its cache is not empty. If the cache is empty too, it will be regarded as end-of-region and open a new scanner for next region.</p>

Description: <p>We have metrics for Get operations at the region server level and region level. </p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

<span class="code-quote">"Get\_num\_ops"</span> : 4837505,

<span class="code-quote">"Get\_min"</span> : 0,

<span class="code-quote">"Get\_max"</span> : 296,

<span class="code-quote">"Get\_mean"</span> : 0.2934618155433431,

<span class="code-quote">"Get\_median"</span> : 0.0,

<span class="code-quote">"Get\_75th\_percentile"</span> : 0.0,

<span class="code-quote">"Get\_95th\_percentile"</span> : 1.0,

<span class="code-quote">"Get\_99th\_percentile"</span> : 1.0,

</pre>

</div></div>

<p>and </p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

<span class="code-quote">"Namespace\_hbase\_table\_meta\_region\_1588230740\_metric\_get\_num\_ops"</span> : 103,

<span class="code-quote">"Namespace\_hbase\_table\_meta\_region\_1588230740\_metric\_get\_min"</span> : 450,

<span class="code-quote">"Namespace\_hbase\_table\_meta\_region\_1588230740\_metric\_get\_max"</span> : 470,

<span class="code-quote">"Namespace\_hbase\_table\_meta\_region\_1588230740\_metric\_get\_mean"</span> : 450.19417475728153,

<span class="code-quote">"Namespace\_hbase\_table\_meta\_region\_1588230740\_metric\_get\_median"</span> : 460.0,

<span class="code-quote">"Namespace\_hbase\_table\_meta\_region\_1588230740\_metric\_get\_75th\_percentile"</span> : 470.0,

<span class="code-quote">"Namespace\_hbase\_table\_meta\_region\_1588230740\_metric\_get\_95th\_percentile"</span> : 470.0,

<span class="code-quote">"Namespace\_hbase\_table\_meta\_region\_1588230740\_metric\_get\_99th\_percentile"</span> : 470.0,

</pre>

</div></div>

<p>The problem is that the report values for the region server shows the latency, versus the reported values for the region shows the response sizes. There is no way of telling this without reading the source code. </p>

<p>I think we should deprecate response size histograms in favor of latency histograms. </p>

<p>See also <a href="https://issues.apache.org/jira/browse/HBASE-15376" title="ScanNext metric is size-based while every other per-operation metric is time based" class="issue-link" data-issue-key="HBASE-15376"><del>HBASE-15376</del></a>. </p>

Description: <p>We have per-operation metrics for <tt>Get</tt>, <tt>Mutate</tt>, <tt>Delete</tt>, <tt>Increment</tt>, and <tt>ScanNext</tt>. </p>

<p>The metrics are emitted like: </p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

<span class="code-quote">"Get\_num\_ops"</span> : 4837505,

<span class="code-quote">"Get\_min"</span> : 0,

<span class="code-quote">"Get\_max"</span> : 296,

<span class="code-quote">"Get\_mean"</span> : 0.2934618155433431,

<span class="code-quote">"Get\_median"</span> : 0.0,

<span class="code-quote">"Get\_75th\_percentile"</span> : 0.0,

<span class="code-quote">"Get\_95th\_percentile"</span> : 1.0,

<span class="code-quote">"Get\_99th\_percentile"</span> : 1.0,

...

<span class="code-quote">"ScanNext\_num\_ops"</span> : 194705,

<span class="code-quote">"ScanNext\_min"</span> : 0,

<span class="code-quote">"ScanNext\_max"</span> : 18441,

<span class="code-quote">"ScanNext\_mean"</span> : 7468.274651395701,

<span class="code-quote">"ScanNext\_median"</span> : 583.0,

<span class="code-quote">"ScanNext\_75th\_percentile"</span> : 583.0,

<span class="code-quote">"ScanNext\_95th\_percentile"</span> : 13481.0,

<span class="code-quote">"ScanNext\_99th\_percentile"</span> : 13481.0,

</pre>

</div></div>

<p>The problem is that all of Get,Mutate,Delete,Increment,Append,Replay are time based tracking how long the operation ran, while ScanNext is tracking returned response sizes (returned cell-sizes to be exact). Obviously, this is very confusing and you would only know this subtlety if you read the metrics collection code. </p>

<p>Not sure how useful is the ScanNext metric as it is today. We can deprecate it, and introduce a time based one to keep track of scan request latencies. </p>

<p>ps. Shamelessly using the parent jira (since these seem relavant). </p>

Description:

Description: <p>I'm used to run unit tests on a remote server before summit a large patch because sometimes a testcase failure can not be reproduced locally.</p>

<p>Recently <a href="https://issues.apache.org/jira/secure/ViewProfile.jspa?name=yangzhe1991" class="user-hover" rel="yangzhe1991">Phil Yang</a> and me share a remote server with a different account. We found that some of the unit tests write to '/tmp' with a fixed name and do not clean the file after test. This cause that the unit test can only be succeeded for one person...</p>

<p>Open an umbrella issue to address this.</p>

Description: <p>I couldn't turn off compaction throughput by following release note instructions. I fixed release notes in parent but also needs code fix in factory class.</p>

Description: <p>Currently, we keep WAL file per table in backup site, this creates significant data duplication in case of many tables in a backup set. We have to keep all WAL files in a single place and keep track of WAL files involved per table/backup id as links (references) </p>

<p>This is not only the data duplication issue, but a performance issue as well (we copy the same file over and over again for every table in a backup set). </p>

Description: <p>In Procedure-V2 Phase 1 (<a href="https://issues.apache.org/jira/browse/HBASE-14336" title="Procedure V2 Phase 1 - Procedure Framework and Making DDL Operations fault tolerant" class="issue-link" data-issue-key="HBASE-14336"><del>HBASE-14336</del></a>), some infrastructure of supporting child procedure exists. However, there is no need in Phase 1 (master DDL) to have multi-level procedures. This JIRA implements adding child procedures to procedure execution list and execute them before parent procedure can make further progress.</p>

Description: <p>MOB feature was integrated to master branch half a year ago.</p>

<p>Since then there have been bug fixes which stabilize the feature.<br/>

Some customers have been using it at PB scale.</p>

<p>Here is discussion thread on dev mailing list:</p>

<p><a href="http://search-hadoop.com/m/YGbbDSqxD1PYXK62/hbase+MOB+in+branch-1&amp;subj=Re+MOB+in+branch+1+Re+RESULT+VOTE+Merge+branch+hbase+11339+HBase+MOB+to+trunk+" class="external-link" rel="nofollow">http://search-hadoop.com/m/YGbbDSqxD1PYXK62/hbase+MOB+in+branch-1&amp;subj=Re+MOB+in+branch+1+Re+RESULT+VOTE+Merge+branch+hbase+11339+HBase+MOB+to+trunk+</a></p>

<p>This issue is to backport MOB feature to branch-1.</p>

Description: <p>If empty or non-existing region name parameter is passed to "region.jsp ?name=" it will cause 500 NPE error. It is not big deal but looks ugly.</p>

Description: <p>To better determine 'hot' data.</p>

Description: <p>DisabledWALProvider is not for production cases. But still configuring one does not allow region opening to work fine and thus hanging create table process.<br/>

[code}<br/>

at org.apache.hadoop.hbase.wal.WALKey.getWriteEntry(WALKey.java:106)<br/>

at org.apache.hadoop.hbase.regionserver.wal.WALUtil.writeMarker(WALUtil.java:133)<br/>

at org.apache.hadoop.hbase.regionserver.wal.WALUtil.writeRegionEventMarker(WALUtil.java:88)<br/>

at org.apache.hadoop.hbase.regionserver.HRegion.writeRegionOpenMarker(HRegion.java:1006)<br/>

at org.apache.hadoop.hbase.regionserver.HRegion.openHRegion(HRegion.java:6483)<br/>

at org.apache.hadoop.hbase.regionserver.HRegion.openHRegion(HRegion.java:6433)<br/>

at org.apache.hadoop.hbase.regionserver.HRegion.openHRegion(HRegion.java:6404)<br/>

at org.apache.hadoop.hbase.regionserver.HRegion.openHRegion(HRegion.java:6360)<br/>

at org.apache.hadoop.hbase.regionserver.HRegion.openHRegion(HRegion.java:6311)<br/>

at org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler.openRegion(OpenRegionHandler.java:282)<br/>

at org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler.process(OpenRegionHandler.java:107)<br/>

at org.apache.hadoop.hbase.executor.EventHandler.run(EventHandler.java:104)<br/>

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)<br/>

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

But there is also a clear comment in WALUtil#writeMarker

</pre>

</div></div>

<p> // If you get hung here, is it a real WAL or a mocked WAL? If the latter, you need to<br/>

// trip the latch that is inside in getWriteEntry up in your mock. See down in the append<br/>

// called from onEvent in FSHLog.</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java"></pre>

</div></div>

<p>But this is about Mocking but here it is not mocking and still we are not able to use it. The only option is like the client should say SKIP\_WAL for every mutation if we really need to avoid WAL.</p>

Description: <p>What hfileblock is doing &#8211; that it overreads when pulling in from hdfs to fetch the header of the next block to save on seeks; that it caches the block and overread and then adds an extra 13 bytes to the cached entry; that buckets in bucketcache have at least four hfileblocks in them and so on &#8211; was totally baffling me. This patch docs the class, adds some trace-level logging so you can see if you are doing the right thing, and then adds a test of file-backed bucketcache that checks that persistence is working.</p>

Description: <p>In testGetPassword, we create a key file at /tmp/foo.jks and set its permissions to 600. This will cause testcase failure on a shared machine.</p>

Description: <p>From <a href="https://builds.apache.org/job/HBase%20Website%20Link%20Ckecker/28/artifact/link\_report/index.html:" class="external-link" rel="nofollow">https://builds.apache.org/job/HBase%20Website%20Link%20Ckecker/28/artifact/link\_report/index.html:</a> </p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

host: hbase.apache.org

date: Mon, 29-Feb-2016 12:06:21 (local)

Linklint version: 2.3.5\_ingo\_020

#------------------------------------------------------------

# warn 2 warnings (cross referenced)

#------------------------------------------------------------

unquoted <span class="code-quote">"&lt;"</span> in &lt;0.90.5, &lt;0.90.5, &lt;

occurred in

/devapidocs/org/apache/hadoop/hbase/util/HBaseFsck.html

unquoted <span class="code-quote">"&lt;"</span> in &lt;0.92.0) a master

res

occurred in

/devapidocs/org/apache/hadoop/hbase/util/HBaseFsck.html

</pre>

</div></div>

Description: <p>There are a number of cases where we can get SASL connection failures before getting to the server, like errors talking to the KDC/TGS and misconfiguration of kerberos principals. Hence these will not show up in the server-side authentication\_failures metric.</p>

<p>We should add client side metrics on SASL connection failures to capture these.</p>

Description: <p>Compression creates conf using new Configuration() and this leads to it not respecting the confs set in hbase-site, fixing it is trivial using HBaseConfiguration.create()</p>

Description: <p>When we were trying to tune the bucket sizes <tt>hbase.bucketcache.bucket.sizes</tt> according to our workload, we encountered an issue due to the way offset is stored in the bucket entry. We divide the offset into integer base and byte value and it assumes that all bucket offsets will be a multiple of 256 (left shifting by 8). See the code below</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

<span class="code-object">long</span> offset() { <span class="code-comment">// Java has no unsigned numbers

</span> <span class="code-object">long</span> o = ((<span class="code-object">long</span>) offsetBase) &amp; 0xFFFFFFFF;

o += (((<span class="code-object">long</span>) (offset1)) &amp; 0xFF) &lt;&lt; 32;

<span class="code-keyword">return</span> o &lt;&lt; 8;

}

<span class="code-keyword">private</span> void setOffset(<span class="code-object">long</span> value) {

<span class="code-keyword">assert</span> (value &amp; 0xFF) == 0;

value &gt;&gt;= 8;

offsetBase = (<span class="code-object">int</span>) value;

offset1 = (<span class="code-object">byte</span>) (value &gt;&gt; 32);

}

</pre>

</div></div>

<p>This was there to save 3 bytes per BucketEntry instead of using long and when there are no other fields in the Bucket Entry, but now there are lot of fields in the bucket entry , This not documented so we could either document the constraint that it should be a strict 256 bytes multiple of just go away with this constraint. </p>

Description: <p>There were several flaky tests added there as part of <a href="https://issues.apache.org/jira/browse/HBASE-15306" title="Make RPC call queue length dynamically configurable" class="issue-link" data-issue-key="HBASE-15306"><del>HBASE-15306</del></a> and likely <a href="https://issues.apache.org/jira/browse/HBASE-15136" title="Explore different queuing behaviors while busy" class="issue-link" data-issue-key="HBASE-15136"><del>HBASE-15136</del></a>.</p>

Description: <p>Now that it is clear that no memstore segment will be implemented as an HFIle, and that all segments store their data in some representation of CellSet (skip-list or flat), the segment hierarchy can be much simplified.</p>

<p>The attached patch includes only 3 classes in the hierarchy:<br/>

Segment - comprises most of the state and implementation<br/>

MutableSegment - extends API with add and rollback functionality<br/>

ImmutableSegment - extends API with key-value scanner for snapshot</p>

<p>SegmentScanner is the scanner for all types of segments. </p>

<p>In addition, the option to rollback immutable segment in the memstore is disabled.</p>

<p>This code would allow us to make progress independently in the compaction subtask (<a href="https://issues.apache.org/jira/browse/HBASE-14920" title="Compacting Memstore" class="issue-link" data-issue-key="HBASE-14920">HBASE-14920</a>) and the flat index representation subtask (<a href="https://issues.apache.org/jira/browse/HBASE-14921" title="Memory optimizations" class="issue-link" data-issue-key="HBASE-14921">HBASE-14921</a>). It also means that the new immutable segment can reuse the existing SegmentScanner, instead of implementing a new scanner.</p>

Description: <p>A small but maybe critical bug</p>

Description: <p>When auto-balance is enabled in TableInputFormatBase and the table key is a binary key, the getSplitKey method does not function correctly for signed bytes. The proposed solution it to utilize org.apache.hadoop.hbase.util.Bytes#split method to find the split key. org.apache.hadoop.hbase.util.Bytes#split is stated to be a expensive operation, so if another solution is preferred, that is fine. In addition, handling of a split key that is equal to the TableSplit end key is added to calculateRebalancedSplits.</p>

Description: <p>Remove unused Imports.</p>

Description: <p>After <a href="https://issues.apache.org/jira/browse/HBASE-10569" class="external-link" rel="nofollow">HBASE-10569</a>, master is also a regionserver and it will serve regions of system tables. The meta region info could be viewed on master at the address such as : <a href="http://localhost:16010/region.jsp?name=1588230740" class="external-link" rel="nofollow">http://localhost:16010/region.jsp?name=1588230740</a>. The real path of region.jsp for the request will be hbase-webapps/master/region.jsp on master, however, the region.jsp is under the directory hbase-webapps/regionserver, so that can not be found on master.</p>

Description: <p>Currently we do not clear/update meta cache for some special exceptions if the operation went through AsyncProcess#submit like HTable#put calls. But, we clear meta cache without checking for these special exceptions in case of other operations like gets, deletes etc because they directly go through the RpcRetryingCaller#callWithRetries instead of the AsyncProcess. </p>

Description: <p>This exception is being thrown more. We should add a metric for this one.</p>

Description: <p>We could improve on the existing <a href="http://hbase.apache.org/devapidocs/org/apache/hadoop/hbase/codec/prefixtree/package-summary.html" class="external-link" rel="nofollow">PREFIX\_TREE block</a> encoder by upgrading the persistent data structure from a trie to a finite state transducer. This would theoretically allow us to reuse bytes not just for rowkey prefixes, but infixes and suffixes as well. My read of the literature means we may also be able to encode values as well, further reducing storage size when values are repeated (ie, a "customer id" field with very low cardinality &#8211; probably happens a lot in our denormalized world). There's a really nice <a href="http://blog.burntsushi.net/transducers/" class="external-link" rel="nofollow">blog post</a> about this data structure, and apparently our siblings in Lucene make heavy use of <a href="http://lucene.apache.org/core/5\_5\_0/core/org/apache/lucene/util/fst/package-summary.html#package\_description" class="external-link" rel="nofollow">their implementation</a>.</p>

Description: <p>As suggested in the parent issue, a fix in the doc needs to be repeated in hbase-default.xml</p>

Description:

Description: <p>So that new properties like surefire.excludesFile and includesFile can be used to easily exclude/include flaky tests.</p>

Description: <p>Counts are appoximate and go away. We should re-work the tests or test utils to make them work now.</p>

Description:

Description:

Description: <p>point hbase-1.3 build to branch-1.3, add hbase-1.4 and point it to branch-1.</p>

Description:

Description:

Description: <p>create branch-1.3 and update branch-1 poms to 1.4.0-SNAPSHOT</p>

Description: <p>Umbrella jira for 1.3 release.</p>

Description: <p>There are cases the region sever will return partial row result, such as the client set batch for scan or configured size limit reached. In these situations, the client may return data that violates the row-level transaction to the application. The following steps show the problem:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

<span class="code-comment">// assume there is a test table 'test\_table' with one family 'F' and one region 'region'.

</span><span class="code-comment">// meanwhile there are two region servers 'rsA' and 'rsB'.

</span>1. Let 'region' firstly located in 'rsA' and put one row with two columns 'c1' and 'c2' as:

&gt; put 'test\_table', 'row', 'F:c1', 'value1', 'F:c2', 'value1'

2. Start a client to scan 'test\_table', with scan.setBatch(1) and scan.setCaching(1). The client will get one column as : {column='F:c1' and value='value1'} in the first rpc call after scanner created, and the result will be returned to application.

3. Before the client issues the next request, the 'region' was moved to 'rsB' and accepted another mutations <span class="code-keyword">for</span> the two columns 'c1' and 'c2' as:

&gt; put 'test\_table', 'row', 'F:c1', 'value2', 'F:c2', 'value2'

4. Then, the client will receive a RegionMovedException when issuing next request and will retry to open scanner on 'rsB'. The newly opened scanner will higher mvcc than old data so that could read out column as : { column='F:c2' with value='value2'} and <span class="code-keyword">return</span> the result to application.

Therefore, the application will get data as:

'row' column='F:c1' value='value1'

'row' column='F:c2', value='value2'

The returned data is combined from two different mutations and violates the row-level transaction.

</pre>

</div></div>

<p>The reason is that the newly opened scanner after region moved will get a different mvcc. I am not sure whether this result is by design for scan if partial row result is allowed. However, such row result combined from different transactions may make the application have unexpected state.</p>

Description: <p>For our MiCloud service, the old data is rarely touched but we still need to keep it, so we want to put the data on inexpensive device and reduce redundancy using EC to cut down the cost.</p>

<p>With date based tiered compaction introduced in <a href="https://issues.apache.org/jira/browse/HBASE-15181" title="A simple implementation of date based tiered compaction" class="issue-link" data-issue-key="HBASE-15181"><del>HBASE-15181</del></a>, new data and old data can be placed in different tier. But the tier boundary moves as time lapse so it is still possible that we do compaction on old tier which breaks our block moving and EC work.</p>

<p>So here we want to introduce an "archive tier" to better fit our scenario. Add an configuration called "archive unit", for example, year. That means, if we find that the tier boundary is already in the previous year, then we reset the boundary to the start of year and end of year, and if we want to do compaction in this tier, just compact all files into one file. The file will never be changed unless we force a major compaction so it is safe to apply EC and other cost reducing approach on the file. And we make more tiers before this tier year by year. </p>

Description: <p>When testing and comparing the performance of different file systems(HDFS, Azure blob storage, AWS S3 and so on) for HBase, it's better to avoid the affect of the HBase BlockCache and get the actually random read latency when data block is read from underlying file system. (Usually, the index block and meta block should be cached in memory in the testing).</p>

<p>So we add a option in CacheConfig to disable the data block cache.</p>

<p>Suggestions are welcomed~ Thanks</p>

Description: <p>We have two new compaction algorithms FIFO and Date tiered that are for time series data. We should document how to use them in the book. </p>

Description: <p>Currently, the connector only support read path. A complete solution should support both read and writer. This subtask add write support.</p>

Description: <p>Add composite key filter support in the connector.</p>

Description: <p>Avro is a popular format for hbase storage. User may want the support natively in the connector.</p>

Description: <p>Currently, the range filter is based on the order of bytes. But for java primitive type, such as short, int, long, double, float, etc, their order is not consistent with their byte order, extra manipulation has to be in place to take care of them correctly.</p>

<p>For example, for the integer range (-100, 100), the filter &lt;= 1, the current filter will return 0 and 1, and the right return value should be (-100, 1]</p>

Description:

Description: <p>The current implementation for restore uses WALReplay M/R job. This has performance and stability problems, since it uses HBase client API to insert data. We have to migrate to bulk load approach: generate hfiles directly from snapshot and incremental images. We run separate M/R job for every backup image between last FULL backup and current incremental backup we restore to and for every table in the list (image). If we have 10 tables and 30 days of incremental backup images - this results in 30x10 = 300 M/R jobs. MUST be optimized.</p>

Description: <p>Currently, we do not track delete/recreate or truncate table events</p>

Description: <p>Minor issue where we write back table name in a few places. Should clean it up:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

} <span class="code-keyword">else</span> {

out.write(<span class="code-quote">"\n &lt;title&gt;Table: "</span>);

out.print( fqtn );

out.write(<span class="code-quote">"&lt;/title&gt;\n "</span>);

}

</pre>

</div></div>

Description: <p>See OWASP page on why we should clean it up someday:</p>

<p><a href="https://www.owasp.org/index.php/Unvalidated\_Redirects\_and\_Forwards\_Cheat\_Sheet" class="external-link" rel="nofollow">https://www.owasp.org/index.php/Unvalidated\_Redirects\_and\_Forwards\_Cheat\_Sheet</a></p>

<p>Here is where we do the redirect:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

@Override

<span class="code-keyword">public</span> void doGet(HttpServletRequest request,

HttpServletResponse response) <span class="code-keyword">throws</span> ServletException, IOException {

<span class="code-object">String</span> redirectUrl = request.getScheme() + <span class="code-quote">":<span class="code-comment">//"</span>

</span> + request.getServerName() + <span class="code-quote">":"</span> + regionServerInfoPort

+ request.getRequestURI();

response.sendRedirect(redirectUrl);

}

}

</pre>

</div></div>

Description: <p>When Canary#writeSniffing is enabled, Canary#checkWriteTableDistribution will make sure the regions of write table distributed on all region servers as:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

<span class="code-object">int</span> numberOfServers = admin.getClusterStatus().getServers().size();

......

<span class="code-object">int</span> numberOfCoveredServers = serverSet.size();

<span class="code-keyword">if</span> (numberOfCoveredServers &lt; numberOfServers) {

admin.balancer();

}

</pre>

</div></div>

<p>The master will also work as a regionserver, so that ClusterStatus#getServers will contain the master. On the other hand, write table of Canary will not be assigned to master, making numberOfCoveredServers always smaller than numberOfServers and admin.balancer always be invoked in each sniffing period. This may cause frequent region moves. A simple fix is excluding master from numberOfServers.</p>

Description: <div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

java.lang.NullPointerException

at org.apache.hadoop.metrics2.lib.MutableHistogram.updateSnapshotMetrics(MutableHistogram.java:72)

at org.apache.hadoop.metrics2.lib.MutableRangeHistogram.snapshot(MutableRangeHistogram.java:59)

at org.apache.hadoop.metrics2.lib.DynamicMetricsRegistry.snapshot(DynamicMetricsRegistry.java:391)

at org.apache.hadoop.hbase.metrics.BaseSourceImpl.getMetrics(BaseSourceImpl.java:146)

at org.apache.hadoop.hbase.test.MetricsAssertHelperImpl.getMetrics(MetricsAssertHelperImpl.java:243)

at org.apache.hadoop.hbase.test.MetricsAssertHelperImpl.getCounter(MetricsAssertHelperImpl.java:201)

at org.apache.hadoop.hbase.regionserver.TestHRegion.testBatchPut\_whileNoRowLocksHeld(TestHRegion.java:1498)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:497)

at org.junit.runners.model.FrameworkMethod$1.runReflectiveCall(FrameworkMethod.java:50)

at org.junit.internal.runners.model.ReflectiveCallable.run(ReflectiveCallable.java:12)

at org.junit.runners.model.FrameworkMethod.invokeExplosively(FrameworkMethod.java:47)

at org.junit.internal.runners.statements.InvokeMethod.evaluate(InvokeMethod.java:17)

at org.junit.internal.runners.statements.RunBefores.evaluate(RunBefores.java:26)

at org.junit.internal.runners.statements.RunAfters.evaluate(RunAfters.java:27)

at org.junit.rules.TestWatcher$1.evaluate(TestWatcher.java:55)

at org.junit.rules.RunRules.evaluate(RunRules.java:20)

at org.junit.runners.ParentRunner.runLeaf(ParentRunner.java:325)

at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:78)

at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:57)

at org.junit.runners.ParentRunner$3.run(ParentRunner.java:290)

at org.junit.runners.ParentRunner$1.schedule(ParentRunner.java:71)

at org.junit.runners.ParentRunner.runChildren(ParentRunner.java:288)

at org.junit.runners.ParentRunner.access$000(ParentRunner.java:58)

at org.junit.runners.ParentRunner$2.evaluate(ParentRunner.java:268)

at org.junit.runners.ParentRunner.run(ParentRunner.java:363)

at org.junit.runner.JUnitCore.run(JUnitCore.java:137)

at com.intellij.junit4.JUnit4IdeaTestRunner.startRunnerWithArgs(JUnit4IdeaTestRunner.java:117)

at com.intellij.rt.execution.junit.JUnitStarter.prepareStreamsAndStart(JUnitStarter.java:234)

at com.intellij.rt.execution.junit.JUnitStarter.main(JUnitStarter.java:74)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:497)

at com.intellij.rt.execution.application.AppMain.main(AppMain.java:144)

</pre>

</div></div>

<p>It seems to be introduced after <a href="https://issues.apache.org/jira/browse/HBASE-15222" title="Use less contended classes for metrics" class="issue-link" data-issue-key="HBASE-15222"><del>HBASE-15222</del></a>, <a href="https://issues.apache.org/jira/secure/ViewProfile.jspa?name=eclark" class="user-hover" rel="eclark">Elliott Clark</a></p>

Description: <p><a href="https://issues.apache.org/jira/browse/HBASE-11544" title="[Ergonomics] hbase.client.scanner.caching is dogged and will try to return batch even if it means OOME" class="issue-link" data-issue-key="HBASE-11544"><del>HBASE-11544</del></a> allow scan rpc return partial of a row to reduce memory usage for one rpc request. And client can setAllowPartial or setBatch to get several cells in a row instead of the whole row.</p>

<p>However, the status of the scanner is saved on server and we need this to get the next part if there is a partial result before. If we move the region to another RS, client will get a NotServingRegionException and open a new scanner to the new RS which will be regarded as a new scan from the end of this row. So the rest cells of the row of last result will be missing.</p>

Description: <p>We introduce jitter for region split decision in <a href="https://issues.apache.org/jira/browse/HBASE-13412" title="Region split decisions should have jitter" class="issue-link" data-issue-key="HBASE-13412"><del>HBASE-13412</del></a>, but the following line in <tt>ConstantSizeRegionSplitPolicy</tt> may cause long value overflow if MAX\_FILESIZE is specified to Long.MAX\_VALUE:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

<span class="code-keyword">this</span>.desiredMaxFileSize += (<span class="code-object">long</span>)(desiredMaxFileSize \* (RANDOM.nextFloat() - 0.5D) \* jitter);

</pre>

</div></div>

<p>In our case we specify MAX\_FILESIZE to Long.MAX\_VALUE to prevent target region to split.</p>

Description: <p>Java CheckAndDelete API accepts Delete object which can be used to delete (a cell / cell version / multiple cells / column family or a row), but the rest api only allows to delete the cell (without any version)</p>

<p>Need to add this capability to rest api.</p>

## Title and Description

Title: [HBASE-15422] Procedure v2 - Avoid double yield

Description: <p>ServerCrashProcedure is using a combination of isYieldBeforeExecuteFromState() and ProcedureYieldException, which may end up in yielding twice. (ServerCrashProcedure is the only user of yield)</p>

Title: [HBASE-15421] Convert TestStoreScanner to junit4 from junit3 and clean up close of scanners

Description: <p>I want to add to this test class but it junit3. Let me convert it over first.</p>

Title: [HBASE-15420] TestCacheConfig failed after HBASE-15338

Description: <p>TestCacheConfig failed after <a href="https://issues.apache.org/jira/browse/HBASE-15338" title="Add a option to disable the data block cache for testing the performance of underlying file system" class="issue-link" data-issue-key="HBASE-15338"><del>HBASE-15338</del></a>.<br/>

Fix it in this issue~</p>

Title: [HBASE-15419] Push HBASE-14850 to apache's git.

Description: <p>Now that Apache's git isn't locked down it's time to push the code where it should be.</p>

Title: [HBASE-15418] Clean up un-used warning in test util

Description:

Title: [HBASE-15417] Calls to ObserverContext#bypass in a region observer's prePut method are inconsistent

Description: <p>Calling ctx.bypass(), where ctx is the ObserverContext object passed in to the region observer's prePut method, results in some inconsistent behavior.</p>

<p>If every other put in the batch is also bypassed, the region observer sees none of these in its postPut method. If there is at least one other put which is not bypassed, the region observer sees all of the puts in the batch <em>including those which were bypassed</em>.</p>

<p>The end result is that, after bypassing a put, that put may or may not end up in the region observer's postPut method. This behavior is dependent solely on which other puts the bypassed put is batched together with.</p>

<p>I tried to find existing tickets for this issue, but was unable to. Apologies if I missed something. The closest issues I could find were <a href="https://issues.apache.org/jira/browse/HBASE-4331" title="Bypassing default actions in prePut fails sometimes with HTable client" class="issue-link" data-issue-key="HBASE-4331"><del>HBASE-4331</del></a> and <a href="https://issues.apache.org/jira/browse/HBASE-11503" title="Inconsistency in CP between HRegion#batchMutate() and MultiRowMutation()" class="issue-link" data-issue-key="HBASE-11503">HBASE-11503</a>, but those didn't seem to quite hit it.</p>

<p>Additionally, I threw together a quick demonstration of this issue: <a href="https://github.com/hwh33/bypass-inconsistency-demo" class="external-link" rel="nofollow">https://github.com/hwh33/bypass-inconsistency-demo</a>. You can run that demo in memory using the testing utility or against a running cluster. I actually haven't had time to test it against a cluster though, so you may encounter bugs if running in that mode (but hopefully not!).</p>

Title: [HBASE-15416] TestHFileBackedByBucketCache is flakey since it went in

Description: <p>Looks like cache content changes during running of test... let me fix. Critical.</p>

Title: [HBASE-15415] Improve Master WebUI snapshot information

Description: <p>On the Master WebUI, we currently show lots of information about the space used by individual snapshots.</p>

<p>We should also give a total space used.</p>

Title: [HBASE-15414] Bound the size of multi request returns and/or allow return of partial result up to client

Description: <p>Some knowledgeable hbase users note that while Scanning now allows you return results in 'chunks' for assembly client-side as a whole result (or the application can see the partials as they come out of the cluster), this ability is absent if you do a multi-get; you might get back more than you bargained for and just as chunking when Scanning makes sense because it makes hbase 'regular', we need the same for multiget.</p>

<p>Parking an issue here for discussion.</p>

Title: [HBASE-15413] Procedure-V2: print out ProcedureInfo during trace

Description: <p>Before the <a href="https://issues.apache.org/jira/browse/HBASE-15100" title="Master WALProcs still never clean up" class="issue-link" data-issue-key="HBASE-15100"><del>HBASE-15100</del></a> refactored the code to print the ProcedureInfo object, we don't have the needs. Now we need to implement a customized toString() function to print out information in the ProcedureInfo object.</p>

Title: [HBASE-15412] Add average region size metric

Description: <p>We have several metrics related to region store file size, num regions, etc per regionserver, but we do not have a single metric to track the average region size per regionserver. </p>

<p>Avg region size is important to look at for deciding on the split policy, etc.</p>

Title: [HBASE-15411] Rewrite backup with Procedure V2

Description: <p>Currently full / incremental backup is driven by BackupHandler (see call() method for flow).</p>

<p>This issue is to rewrite the flow using Procedure V2.</p>

<p>States (enum) for full / incremental backup would be introduced in Backup.proto which correspond to the steps performed in BackupHandler#call().<br/>

executeFromState() would pace the backup based on the current state.<br/>

serializeStateData() / deserializeStateData() would be used to persist state into procedure WAL.</p>

Title: [HBASE-15410] Utilize the max seek value when all Filters in MUST\_PASS\_ALL FilterList return SEEK\_NEXT\_USING\_HINT

Description: <p>As Preston mentioned in the comment in <a href="https://issues.apache.org/jira/browse/HBASE-15243" title="Utilize the lowest seek value when all Filters in MUST\_PASS\_ONE FilterList return SEEK\_NEXT\_USING\_HINT" class="issue-link" data-issue-key="HBASE-15243"><del>HBASE-15243</del></a>:</p>

<p><a href="https://issues.apache.org/jira/browse/HBASE-15243?focusedCommentId=15143557&amp;page=com.atlassian.jira.plugin.system.issuetabpanels:comment-tabpanel#comment-15143557" class="external-link" rel="nofollow">https://issues.apache.org/jira/browse/HBASE-15243?focusedCommentId=15143557&amp;page=com.atlassian.jira.plugin.system.issuetabpanels:comment-tabpanel#comment-15143557</a></p>

<p>an optimization for filters returning a SEEK\_NEXT\_USING\_HINT would be to seek to the highest hint (Any previous/lower row won't be accepted by the filter returning that seek).</p>

<p>This JIRA is to explore this potential optimization.</p>

Title: [HBASE-15409] TestHFileBackedByBucketCache failed randomly on jdk8

Description: <p>When running the small tests, we found TestHFileBackedByBucketCache failed randomly</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

mvn clean <span class="code-keyword">package</span> install -DrunSmallTests -Dtest=TestHFileBackedByBucketCache

Running org.apache.hadoop.hbase.io.hfile.TestHFileBackedByBucketCache

Tests run: 1, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 1.262 sec &lt;&lt;&lt; FAILURE! - in org.apache.hadoop.hbase.io.hfile.TestHFileBackedByBucketCache

testBucketCacheCachesAndPersists(org.apache.hadoop.hbase.io.hfile.TestHFileBackedByBucketCache) Time elapsed: 0.69 sec &lt;&lt;&lt; FAILURE!

java.lang.AssertionError: expected:&lt;5&gt; but was:&lt;4&gt;

at org.apache.hadoop.hbase.io.hfile.TestHFileBackedByBucketCache.testBucketCacheCachesAndPersists(TestHFileBackedByBucketCache.java:161)

</pre>

</div></div>

Title: [HBASE-15408] MiniCluster's master crashes and unit tests timeout

Description: <p>These days there are many tests timeout on builds.apache.org. I have no log on timeout tests but I find a possible reason: master crashes and MiniCluster will log "No master found; retry" forever until timeout.</p>

Title: [HBASE-15407] Add SASL support for fan out OutputStream

Description: <p>Otherwise we can not use it in secure environment.</p>

<p>Should be a netty handler, but see</p>

<p><a href="https://github.com/netty/netty/issues/1966" class="external-link" rel="nofollow">https://github.com/netty/netty/issues/1966</a></p>

<p>I do not think it will be available in the near future, so we need to do it by ourselves.</p>

Title: [HBASE-15406] Split / merge switch left disabled after early termination of hbck

Description: <p>This was what I did on cluster with 1.4.0-SNAPSHOT built Thursday:</p>

<p>Run 'hbase hbck -disableSplitAndMerge' on gateway node of the cluster<br/>

Terminate hbck early<br/>

Enter hbase shell where I observed:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

hbase(main):001:0&gt; splitormerge\_enabled 'SPLIT'

<span class="code-keyword">false</span>

0 row(s) in 0.3280 seconds

hbase(main):002:0&gt; splitormerge\_enabled 'MERGE'

<span class="code-keyword">false</span>

0 row(s) in 0.0070 seconds

</pre>

</div></div>

<p>Expectation is that the split / merge switches should be restored to default value after hbck exits.</p>

Title: [HBASE-15405] Fix PE logging and wrong defaults in help message

Description: <p>Corrects wrong default values for few options in the help message.</p>

<p>Final stats from multiple clients are intermingled making it hard to understand. Also the logged stats aren't very machine readable. It can be helpful in a daily perf testing rig which scraps logs for results.</p>

<p>Example of logs before the change.</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest latency log (microseconds), on 1048576 measures

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest latency log (microseconds), on 1048576 measures

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest latency log (microseconds), on 1048576 measures

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest latency log (microseconds), on 1048576 measures

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest latency log (microseconds), on 1048576 measures

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest latency log (microseconds), on 1048576 measures

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest latency log (microseconds), on 1048576 measures

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: 0/1048570/1048576, latency mean=953.98, min=359.00, max=324050.00, stdDev=851.82, 95th=1368.00, 99th=1625.00

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: 0/1048570/1048576, latency mean=953.92, min=356.00, max=323394.00, stdDev=817.55, 95th=1370.00, 99th=1618.00

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: 0/1048570/1048576, latency mean=953.98, min=367.00, max=322745.00, stdDev=840.43, 95th=1369.00, 99th=1622.00

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest latency log (microseconds), on 1048576 measures

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest latency log (microseconds), on 1048576 measures

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest latency log (microseconds), on 1048576 measures

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest Min = 375.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest Min = 363.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest Avg = 953.6624126434326

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest Avg = 953.4124526977539

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest StdDev = 781.3929776087633

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest StdDev = 742.8027916717297

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 50th = 894.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 50th = 894.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 75th = 1070.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 75th = 1071.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 95th = 1369.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 95th = 1369.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 99th = 1623.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 99th = 1624.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest Min = 372.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 99.9th = 3013.9980000000214

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest Avg = 953.2451229095459

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 99.9th = 3043.9980000000214

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest StdDev = 725.4744472152282

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 99.99th = 25282.380199996755

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 50th = 895.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 99.99th = 25812.763399994

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 75th = 1071.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 99.999th = 89772.78990004538

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 95th = 1369.0

16/03/05 22:43:06 INFO hbase.PerformanceEvaluation: IncrementTest 99.999th = 122808.39587019826

</pre>

</div></div>

<p>After the change</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>16/03/06 02:38:21 INFO hbase.PerformanceEvaluation: Test : RandomWriteTest, Thread : TestClient-1

16/03/06 02:38:21 INFO hbase.PerformanceEvaluation: Latency (us) : , mean=6.23, min=2.00, max=101433.00, stdDev=246.62, 50th=2.00, 75th=2.00, 95th=3.00, 99th=13.00, 99.9th=558.00, 99.99th=9656.19, 99.999th=20213.63

16/03/06 02:38:21 INFO hbase.PerformanceEvaluation: Num measures (latency) : 1048576

16/03/06 02:38:21 INFO hbase.PerformanceEvaluation: ValueSize (bytes) : , mean=0.00, min=0.00, max=0.00, stdDev=0.00, 50th=0.00, 75th=0.00, 95th=0.00, 99th=0.00, 99.9th=0.00, 99.99th=0.00, 99.999th=0.00

16/03/06 02:38:21 INFO hbase.PerformanceEvaluation: Num measures (ValueSize): 0

...

16/03/06 02:38:21 INFO hbase.PerformanceEvaluation: Test : RandomWriteTest, Thread : TestClient-0

16/03/06 02:38:21 INFO hbase.PerformanceEvaluation: Latency (us) : , mean=6.23, min=1.00, max=99321.00, stdDev=246.41, 50th=2.00, 75th=2.00, 95th=3.00, 99th=13.00, 99.9th=566.00, 99.99th=9694.60, 99.999th=23311.86

....

</pre>

</div></div>

Title: [HBASE-15404] PE: Clients in append and increment are operating serially

Description: <p>On running hbase pe --nomapred increment/append 10, i see the following output where it seems like threads are executing operations serially. In the UI too, only one RS is getting requests at a time.</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>6/03/05 22:26:15 INFO hbase.PerformanceEvaluation: Timed test starting in thread TestClient-1

16/03/05 22:26:15 INFO hbase.PerformanceEvaluation: Timed test starting in thread TestClient-2

16/03/05 22:26:15 INFO hbase.PerformanceEvaluation: Timed test starting in thread TestClient-8

16/03/05 22:26:15 INFO hbase.PerformanceEvaluation: Timed test starting in thread TestClient-0

16/03/05 22:26:15 INFO hbase.PerformanceEvaluation: Timed test starting in thread TestClient-4

16/03/05 22:26:15 INFO hbase.PerformanceEvaluation: Timed test starting in thread TestClient-6

16/03/05 22:26:15 INFO hbase.PerformanceEvaluation: Timed test starting in thread TestClient-7

16/03/05 22:26:15 INFO hbase.PerformanceEvaluation: Timed test starting in thread TestClient-5

16/03/05 22:26:15 INFO hbase.PerformanceEvaluation: Timed test starting in thread TestClient-9

16/03/05 22:26:15 INFO hbase.PerformanceEvaluation: Timed test starting in thread TestClient-3

16/03/05 22:27:54 INFO hbase.PerformanceEvaluation: 0/104857/1048576, latency mean=942.48, min=390.00, max=163444.00, stdDev=892.64, 95th=1361.00, 99th=1605.00

16/03/05 22:27:54 INFO hbase.PerformanceEvaluation: 0/104857/1048576, latency mean=942.53, min=366.00, max=163400.00, stdDev=885.49, 95th=1361.00, 99th=1605.00

16/03/05 22:27:54 INFO hbase.PerformanceEvaluation: 0/104857/1048576, latency mean=942.41, min=402.00, max=163436.00, stdDev=891.54, 95th=1359.00, 99th=1602.41

16/03/05 22:27:54 INFO hbase.PerformanceEvaluation: 0/104857/1048576, latency mean=942.51, min=399.00, max=163610.00, stdDev=892.40, 95th=1360.00, 99th=1600.00

16/03/05 22:27:54 INFO hbase.PerformanceEvaluation: 0/104857/1048576, latency mean=942.59, min=393.00, max=162932.00, stdDev=887.65, 95th=1361.00, 99th=1604.00

16/03/05 22:27:54 INFO hbase.PerformanceEvaluation: 0/104857/1048576, latency mean=942.26, min=385.00, max=163482.00, stdDev=891.71, 95th=1358.00, 99th=1599.00

16/03/05 22:27:54 INFO hbase.PerformanceEvaluation: 0/104857/1048576, latency mean=942.51, min=383.00, max=163246.00, stdDev=888.07, 95th=1360.00, 99th=1605.00

16/03/05 22:27:54 INFO hbase.PerformanceEvaluation: 0/104857/1048576, latency mean=942.45, min=385.00, max=163405.00, stdDev=886.65, 95th=1359.00, 99th=1604.00

16/03/05 22:27:54 INFO hbase.PerformanceEvaluation: 0/104857/1048576, latency mean=942.38, min=400.00, max=163580.00, stdDev=887.28, 95th=1359.00, 99th=1602.00

16/03/05 22:27:54 INFO hbase.PerformanceEvaluation: 0/104857/1048576, latency mean=942.29, min=407.00, max=163403.00, stdDev=889.77, 95th=1357.00, 99th=1597.00

16/03/05 22:29:35 INFO hbase.PerformanceEvaluation: 0/209714/1048576, latency mean=950.75, min=366.00, max=163400.00, stdDev=817.84, 95th=1363.00, 99th=1605.00

16/03/05 22:29:35 INFO hbase.PerformanceEvaluation: 0/209714/1048576, latency mean=950.75, min=383.00, max=163246.00, stdDev=821.95, 95th=1363.00, 99th=1604.00

16/03/05 22:29:35 INFO hbase.PerformanceEvaluation: 0/209714/1048576, latency mean=950.75, min=389.00, max=163444.00, stdDev=824.03, 95th=1364.00, 99th=1603.00

16/03/05 22:29:35 INFO hbase.PerformanceEvaluation: 0/209714/1048576, latency mean=950.56, min=382.00, max=163403.00, stdDev=822.44, 95th=1363.00, 99th=1603.00

16/03/05 22:29:35 INFO hbase.PerformanceEvaluation: 0/209714/1048576, latency mean=950.79, min=393.00, max=162932.00, stdDev=818.75, 95th=1365.00, 99th=1601.84

16/03/05 22:29:35 INFO hbase.PerformanceEvaluation: 0/209714/1048576, latency mean=950.70, min=388.00, max=163436.00, stdDev=823.52, 95th=1364.00, 99th=1606.00

16/03/05 22:29:35 INFO hbase.PerformanceEvaluation: 0/209714/1048576, latency mean=950.72, min=376.00, max=163405.00, stdDev=820.65, 95th=1364.00, 99th=1605.00

16/03/05 22:29:35 INFO hbase.PerformanceEvaluation: 0/209714/1048576, latency mean=950.56, min=382.00, max=163482.00, stdDev=823.43, 95th=1363.00, 99th=1599.00

16/03/05 22:29:35 INFO hbase.PerformanceEvaluation: 0/209714/1048576, latency mean=950.67, min=376.00, max=163580.00, stdDev=821.59, 95th=1364.00, 99th=1602.00

16/03/05 22:29:35 INFO hbase.PerformanceEvaluation: 0/209714/1048576, latency mean=950.77, min=390.00, max=163610.00, stdDev=823.88, 95th=1363.00, 99th=1600.00

16/03/05 22:31:16 INFO hbase.PerformanceEvaluation: 0/314571/1048576, latency mean=952.21, min=369.00, max=162932.00, stdDev=787.36, 95th=1361.00, 99th=1595.00

16/03/05 22:31:16 INFO hbase.PerformanceEvaluation: 0/314571/1048576, latency mean=952.18, min=374.00, max=163610.00, stdDev=791.11, 95th=1359.00, 99th=1594.00

16/03/05 22:31:16 INFO hbase.PerformanceEvaluation: 0/314571/1048576, latency mean=952.30, min=367.00, max=163444.00, stdDev=802.21, 95th=1362.00, 99th=1597.00

16/03/05 22:31:16 INFO hbase.PerformanceEvaluation: 0/314571/1048576, latency mean=952.38, min=366.00, max=163400.00, stdDev=799.61, 95th=1360.00, 99th=1596.00

16/03/05 22:31:16 INFO hbase.PerformanceEvaluation: 0/314571/1048576, latency mean=952.31, min=375.00, max=163580.00, stdDev=802.77, 95th=1359.00, 99th=1596.00

16/03/05 22:31:16 INFO hbase.PerformanceEvaluation: 0/314571/1048576, latency mean=952.12, min=388.00, max=163436.00, stdDev=791.34, 95th=1361.00, 99th=1598.00

16/03/05 22:31:16 INFO hbase.PerformanceEvaluation: 0/314571/1048576, latency mean=952.13, min=368.00, max=163405.00, stdDev=788.29, 95th=1360.00, 99th=1598.00

16/03/05 22:31:16 INFO hbase.PerformanceEvaluation: 0/314571/1048576, latency mean=952.35, min=383.00, max=163246.00, stdDev=801.26, 95th=1362.00, 99th=1599.00

16/03/05 22:31:16 INFO hbase.PerformanceEvaluation: 0/314571/1048576, latency mean=952.21, min=382.00, max=163403.00, stdDev=801.88, 95th=1359.00, 99th=1598.00

</pre>

</div></div>

Title: [HBASE-15403] Performance Evaluation tool isn't working as expected

Description: <p>hbase pe --nomapred --rows=100 --table='t4' randomWrite 10</p>

<ol>

<li>count on t4 gives 620 rows</li>

</ol>

<p>hbase pe --nomapred --rows=200 --table='t5' randomWrite 10</p>

<ol>

<li>count on t5 gives 1257 rows</li>

</ol>

<p>hbase pe --nomapred --table='t6' --rows=200 randomWrite 1</p>

<ol>

<li>count on t6 gives 126 rows</li>

</ol>

<p>I was working with 1.2.0, but it's likely that it'll also be affecting master.</p>

Title: [HBASE-15402] Add on thrift server that uses cpp client.

Description:

Title: [HBASE-15401] Update third-party deps and add Zoopeeker

Description:

Title: [HBASE-15400] Use DateTieredCompactor for Date Tiered Compaction

Description: <p>When we compact, we can output multiple files along the current window boundaries. There are two use cases:</p>

<p>1. Major compaction: We want to output date tiered store files with data older than max age archived in trunks of the window size on the higher tier.<br/>

2. Bulk load files and the old file generated by major compaction before upgrading to DTCP.</p>

<p>Pros: <br/>

1. Restore locality, process versioning, updates and deletes while maintaining the tiered layout.<br/>

2. The best way to fix a skewed layout.</p>

<p>This work is based on a prototype of DateTieredCompactor from <a href="https://issues.apache.org/jira/browse/HBASE-15389" title="Write out multiple files when compaction" class="issue-link" data-issue-key="HBASE-15389">HBASE-15389</a> and focused on the part to meet needs for these two use cases while supporting others. I have to call out a few design decisions:</p>

<p>1. We only want to output the files along all windows for major compaction. And we want to output multiple files older than max age in the trunks of the maximum tier window size determined by base window size, windows per tier and max age.</p>

<p>2. For minor compaction, we don't want to output too many files, which will remain around because of current restriction of contiguous compaction by seq id. I will only output two files if all the files in the windows are being combined, one for the data within window and the other for the out-of-window tail. If there is any file in the window excluded from compaction, only one file will be output from compaction. When the windows are promoted, the situation of out of order data will gradually improve. For the incoming window, we need to accommodate the case with user-specified future data.</p>

<p>3. We have to pass the boundaries with the list of store file as a complete time snapshot instead of two separate calls because window layout is determined by the time the computation is called. So we will need new type of compaction request. </p>

<p>4. Since we will assign the same seq id for all output files, we need to sort by maxTimestamp subsequently. Right now all compaction policy gets the files sorted for StoreFileManager which sorts by seq id and other criteria. I will use this order for DTCP only, to avoid impacting other compaction policies. </p>

<p>5. We need some cleanup of current design of StoreEngine and CompactionPolicy.</p>

Title: [HBASE-15399] JavaHBaseContext is missung the BulkLoad calls.

Description: <p>Sound like bulkLoad methods are there only on the Scala side. Doesn't seems to be doable on the Java side. Need to add it into JavaHBaseContext.</p>

Title: [HBASE-15398] Cells loss or disorder when using family essential filter and partial scanning protocol

Description: <p>In RegionScannerImpl, we have two heaps, storeHeap and joinedHeap. If we have a filter and it doesn't apply to all cf, the stores whose families needn't be filtered will be in joinedHeap. We scan storeHeap first, then joinedHeap, and merge the results and sort and return to client. We need sort because the order of Cell is rowkey/cf/cq/ts and a smaller cf may be in the joinedHeap.</p>

<p>However, after <a href="https://issues.apache.org/jira/browse/HBASE-11544" title="[Ergonomics] hbase.client.scanner.caching is dogged and will try to return batch even if it means OOME" class="issue-link" data-issue-key="HBASE-11544"><del>HBASE-11544</del></a> we may transfer partial results when we get SIZE\_LIMIT\_REACHED\_MID\_ROW or other similar states. We may return a larger cf first because it is in storeHeap and then a smaller cf because it is in joinedHeap. Server won't hold all cells in a row and client doesn't have a sorting logic. The order of cf in Result for user is wrong.</p>

<p>And a more critical bug is, if we get a LIMIT\_REACHED\_MID\_ROW on the last cell of a row in storeHeap, we will break scanning in RegionScannerImpl and in populateResult we will change the state to SIZE\_LIMIT\_REACHED because next peeked cell is next row. But this is only the last cell of one and we have two... And SIZE\_LIMIT\_REACHED means this Result is not partial (by ScannerContext.partialResultFormed), client will see it and merge them and return to user with losing data of joinedHeap. On next scan we will read next row of storeHeap and joinedHeap is forgotten and never be read...</p>

Title: [HBASE-15397] Create bulk load replication znode(hfile-refs) in ZK replication queue by default

Description: <p>Create bulk load replication znode(hfile-refs) in ZK replication queue by default same as hbase replication znode. <br/>

Otherwise the problem what happens is currently replication admin directly operates on ZK without routing through HM/RS. So suppose if a user enables the replication for bulk loaded data in server but fails to do the same in the client configurations then add peer will not add hfile-refs znode, resulting in replication failure for bulk loaded data.<br/>

So after fixing this the behavior will be same as mutation replication.</p>

Title: [HBASE-15396] Enhance mapreduce.TableSplit to add encoded region name

Description: <p>When troubleshooting HBase-input MR job map failures, the TableSplit's toString() definition comes very handy. However, it still does not tell us directly which region ID it was created for (it does tell us the key and the host, but not the exact region name, so there's a second lookup needed from the UI to find the ID to then check the servers with).</p>

<p>It would be beneficial to have the encoded region name as part of the output, so this extra manual lookup can be avoided. The encoded name is much easier to find events with.</p>

Title: [HBASE-15395] HMaster ui sometimes doesn't display anything under user/system tables

Description: <p>tried reloading the page and clicking on the tables, shows nothing, though regions are assigned/</p>

Title: [HBASE-15394] Orphan daughter region directories are not cleared which have been left around from previous split attempt

Description: <p>This is the case where daughter region A is created successfully under table dir but region server crashed during creating daughter region B. <br/>

So split failed and left an orphan daughter dir in the filesystem.</p>

<p>Currently we are just clearing the ".split" directory not the orphan daughter regions which might be moved out under the table directory during previous failed SPLIT operation.</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

void cleanupAnySplitDetritus() <span class="code-keyword">throws</span> IOException {

Path splitdir = <span class="code-keyword">this</span>.getSplitsDir();

<span class="code-keyword">if</span> (!fs.exists(splitdir)) <span class="code-keyword">return</span>;

<span class="code-comment">// Look at the splitdir. It could have the encoded names of the daughter

</span> <span class="code-comment">// regions we tried to make. See <span class="code-keyword">if</span> the daughter regions actually got made

</span> <span class="code-comment">// out under the tabledir. If here under splitdir still, then the split did

</span> <span class="code-comment">// not complete. Try and <span class="code-keyword">do</span> cleanup. This code WILL NOT <span class="code-keyword">catch</span> the <span class="code-keyword">case</span>

</span> <span class="code-comment">// where we successfully created daughter a but regionserver crashed during

</span> <span class="code-comment">// the creation of region b. In <span class="code-keyword">this</span> <span class="code-keyword">case</span>, there'll be an orphan daughter

</span> <span class="code-comment">// dir in the filesystem. TOOD: Fix.

</span> FileStatus[] daughters = FSUtils.listStatus(fs, splitdir, <span class="code-keyword">new</span> FSUtils.DirFilter(fs));

<span class="code-keyword">if</span> (daughters != <span class="code-keyword">null</span>) {

<span class="code-keyword">for</span> (FileStatus daughter: daughters) {

Path daughterDir = <span class="code-keyword">new</span> Path(getTableDir(), daughter.getPath().getName());

<span class="code-keyword">if</span> (fs.exists(daughterDir) &amp;&amp; !deleteDir(daughterDir)) {

<span class="code-keyword">throw</span> <span class="code-keyword">new</span> IOException(<span class="code-quote">"Failed delete of "</span> + daughterDir);

}

}

}

cleanupSplitsDir();

LOG.info(<span class="code-quote">"Cleaned up old failed split transaction detritus: "</span> + splitdir);

}

</pre>

</div></div>

<p>Creating this JIRA is to track these changes.</p>

Title: [HBASE-15393] Enable table replication command will fail when parent znode is not default in peer cluster

Description: <p>Enable table replication command will fail when parent znode is not /hbase(default) in peer cluster and there is only one peer cluster added in the source cluster.</p>

Title: [HBASE-15392] Single Cell Get reads two HFileBlocks

Description: <p>As found by Daniel "SystemTap" Pol, a simple Get results in our reading two HFileBlocks, the one that contains the wanted Cell, and the block that follows.</p>

<p>Here is a bit of custom logging that logs a stack trace on each HFileBlock read so you can see the call stack responsible:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

2016-03-03 22:20:30,191 INFO [B.defaultRpcServer.handler=20,queue=2,port=16020] regionserver.StoreScanner: START LOOP

2016-03-03 22:20:30,192 INFO [B.defaultRpcServer.handler=20,queue=2,port=16020] regionserver.StoreScanner: QCODE SEEK\_NEXT\_COL

2016-03-03 22:20:30,192 INFO [B.defaultRpcServer.handler=20,queue=2,port=16020] hfile.HFileBlockIndex: STARTED WHILE

2016-03-03 22:20:30,192 INFO [B.defaultRpcServer.handler=20,queue=2,port=16020] hfile.CombinedBlockCache: OUT OF L2

2016-03-03 22:20:30,192 TRACE [B.defaultRpcServer.handler=20,queue=2,port=16020] bucket.BucketCache: Read offset=31409152, len=2103

2016-03-03 22:20:30,192 TRACE [B.defaultRpcServer.handler=20,queue=2,port=16020] bucket.FileIOEngine: offset=31409152, length=2103

2016-03-03 22:20:30,193 TRACE [B.defaultRpcServer.handler=20,queue=2,port=16020] hfile.HFileReaderImpl: From Cache [blockType=DATA, fileOffset=2055421, headerSize=33, onDiskSizeWithoutHeader=2024, uncompressedSizeWithoutHeader=2020, prevBlockOffset=2053364, isUseHBaseChecksum=<span class="code-keyword">true</span>, checksumType=CRC32C, bytesPerChecksum=16384, onDiskDataSizeWithHeader=2053, getOnDiskSizeWithHeader=2057, totalChecksumBytes=4, isUnpacked=<span class="code-keyword">true</span>, buf=[org.apache.hadoop.hbase.nio.SingleByteBuff@e19fbd54], dataBeginsWith=\x00\x00\x00)\x00\x00\x01`\x00\x16user995139035672819231, fileContext=[usesHBaseChecksum=<span class="code-keyword">true</span>, checksumType=CRC32C, bytesPerChecksum=16384, blocksize=65536, encoding=NONE, includesMvcc=<span class="code-keyword">true</span>, includesTags=<span class="code-keyword">false</span>, compressAlgo=NONE, compressTags=<span class="code-keyword">false</span>, cryptoContext=[cipher=NONE keyHash=NONE]]]

2016-03-03 22:20:30,193 TRACE [B.defaultRpcServer.handler=20,queue=2,port=16020] hfile.HFileReaderImpl: Cache hit <span class="code-keyword">return</span> [blockType=DATA, fileOffset=2055421, headerSize=33, onDiskSizeWithoutHeader=2024, uncompressedSizeWithoutHeader=2020, prevBlockOffset=2053364, isUseHBaseChecksum=<span class="code-keyword">true</span>, checksumType=CRC32C, bytesPerChecksum=16384, onDiskDataSizeWithHeader=2053, getOnDiskSizeWithHeader=2057, totalChecksumBytes=4, isUnpacked=<span class="code-keyword">true</span>, buf=[org.apache.hadoop.hbase.nio.SingleByteBuff@e19fbd54], dataBeginsWith=\x00\x00\x00)\x00\x00\x01`\x00\x16user995139035672819231, fileContext=[usesHBaseChecksum=<span class="code-keyword">true</span>, checksumType=CRC32C, bytesPerChecksum=16384, blocksize=65536, encoding=NONE, includesMvcc=<span class="code-keyword">true</span>, includesTags=<span class="code-keyword">false</span>, compressAlgo=NONE, compressTags=<span class="code-keyword">false</span>, cryptoContext=[cipher=NONE keyHash=NONE]]]

java.lang.Throwable

at org.apache.hadoop.hbase.io.hfile.HFileReaderImpl.readBlock(HFileReaderImpl.java:1515)

at org.apache.hadoop.hbase.io.hfile.HFileBlockIndex$CellBasedKeyBlockIndexReader.loadDataBlockWithScanInfo(HFileBlockIndex.java:324)

at org.apache.hadoop.hbase.io.hfile.HFileReaderImpl$HFileScannerImpl.seekTo(HFileReaderImpl.java:831)

at org.apache.hadoop.hbase.io.hfile.HFileReaderImpl$HFileScannerImpl.reseekTo(HFileReaderImpl.java:812)

at org.apache.hadoop.hbase.regionserver.StoreFileScanner.reseekAtOrAfter(StoreFileScanner.java:288)

at org.apache.hadoop.hbase.regionserver.StoreFileScanner.reseek(StoreFileScanner.java:198)

at org.apache.hadoop.hbase.regionserver.NonLazyKeyValueScanner.doRealSeek(NonLazyKeyValueScanner.java:54)

at org.apache.hadoop.hbase.regionserver.KeyValueHeap.generalizedSeek(KeyValueHeap.java:321)

at org.apache.hadoop.hbase.regionserver.KeyValueHeap.requestSeek(KeyValueHeap.java:279)

at org.apache.hadoop.hbase.regionserver.StoreScanner.reseek(StoreScanner.java:806)

at org.apache.hadoop.hbase.regionserver.StoreScanner.seekAsDirection(StoreScanner.java:795)

at org.apache.hadoop.hbase.regionserver.StoreScanner.next(StoreScanner.java:624)

at org.apache.hadoop.hbase.regionserver.KeyValueHeap.next(KeyValueHeap.java:153)

at org.apache.hadoop.hbase.regionserver.HRegion$RegionScannerImpl.populateResult(HRegion.java:5703)

at org.apache.hadoop.hbase.regionserver.HRegion$RegionScannerImpl.nextInternal(HRegion.java:5849)

at org.apache.hadoop.hbase.regionserver.HRegion$RegionScannerImpl.nextRaw(HRegion.java:5622)

at org.apache.hadoop.hbase.regionserver.HRegion$RegionScannerImpl.next(HRegion.java:5598)

at org.apache.hadoop.hbase.regionserver.HRegion$RegionScannerImpl.next(HRegion.java:5584)

at org.apache.hadoop.hbase.regionserver.RSRpcServices.get(RSRpcServices.java:2187)

at org.apache.hadoop.hbase.regionserver.RSRpcServices.get(RSRpcServices.java:2123)

at org.apache.hadoop.hbase.protobuf.generated.ClientProtos$ClientService$2.callBlockingMethod(ClientProtos.java:33512)

at org.apache.hadoop.hbase.ipc.RpcServer.call(RpcServer.java:2229)

at org.apache.hadoop.hbase.ipc.CallRunner.run(CallRunner.java:109)

at org.apache.hadoop.hbase.ipc.RpcExecutor.consumerLoop(RpcExecutor.java:136)

at org.apache.hadoop.hbase.ipc.RpcExecutor$1.run(RpcExecutor.java:111)

at java.lang.<span class="code-object">Thread</span>.run(<span class="code-object">Thread</span>.java:745)

2016-03-03 22:20:30,193 INFO [B.defaultRpcServer.handler=20,queue=2,port=16020] hfile.HFileBlockIndex: READ [blockType=DATA, fileOffset=2055421, headerSize=33, onDiskSizeWithoutHeader=2024, uncompressedSizeWithoutHeader=2020, prevBlockOffset=2053364, isUseHBaseChecksum=<span class="code-keyword">true</span>, checksumType=CRC32C, bytesPerChecksum=16384, onDiskDataSizeWithHeader=2053, getOnDiskSizeWithHeader=2057, totalChecksumBytes=4, isUnpacked=<span class="code-keyword">true</span>, buf=[org.apache.hadoop.hbase.nio.SingleByteBuff@e19fbd54], dataBeginsWith=\x00\x00\x00)\x00\x00\x01`\x00\x16user995139035672819231, fileContext=[usesHBaseChecksum=<span class="code-keyword">true</span>, checksumType=CRC32C, bytesPerChecksum=16384, blocksize=65536, encoding=NONE, includesMvcc=<span class="code-keyword">true</span>, includesTags=<span class="code-keyword">false</span>, compressAlgo=NONE, compressTags=<span class="code-keyword">false</span>, cryptoContext=[cipher=NONE keyHash=NONE]]]

2016-03-03 22:20:30,193 INFO [B.defaultRpcServer.handler=20,queue=2,port=16020] hfile.HFileBlockIndex: DATA BLOCK IS TRUE BREAK

2016-03-03 22:20:30,193 INFO [B.defaultRpcServer.handler=20,queue=2,port=16020] hfile.HFileBlockIndex: RETURNING

2016-03-03 22:20:30,194 INFO [B.defaultRpcServer.handler=20,queue=2,port=16020] regionserver.StoreScanner: QCODE INCLUDE

2016-03-03 22:20:30,194 INFO [B.defaultRpcServer.handler=20,queue=2,port=16020] regionserver.StoreScanner: QCODE SKIP

2016-03-03 22:20:30,194 INFO [B.defaultRpcServer.handler=20,queue=2,port=16020] hfile.CombinedBlockCache: OUT OF L2

2016-03-03 22:20:30,194 TRACE [B.defaultRpcServer.handler=20,queue=2,port=16020] bucket.BucketCache: Read offset=30932992, len=2103

2016-03-03 22:20:30,194 TRACE [B.defaultRpcServer.handler=20,queue=2,port=16020] bucket.FileIOEngine: offset=30932992, length=2103

2016-03-03 22:20:30,194 TRACE [B.defaultRpcServer.handler=20,queue=2,port=16020] hfile.HFileReaderImpl: From Cache [blockType=DATA, fileOffset=2057478, headerSize=33, onDiskSizeWithoutHeader=2024, uncompressedSizeWithoutHeader=2020, prevBlockOffset=2055421, isUseHBaseChecksum=<span class="code-keyword">true</span>, checksumType=CRC32C, bytesPerChecksum=16384, onDiskDataSizeWithHeader=2053, getOnDiskSizeWithHeader=2057, totalChecksumBytes=4, isUnpacked=<span class="code-keyword">true</span>, buf=[org.apache.hadoop.hbase.nio.SingleByteBuff@6063ac95], dataBeginsWith=\x00\x00\x00)\x00\x00\x01`\x00\x16user995698996184959679, fileContext=[usesHBaseChecksum=<span class="code-keyword">true</span>, checksumType=CRC32C, bytesPerChecksum=16384, blocksize=65536, encoding=NONE, includesMvcc=<span class="code-keyword">true</span>, includesTags=<span class="code-keyword">false</span>, compressAlgo=NONE, compressTags=<span class="code-keyword">false</span>, cryptoContext=[cipher=NONE keyHash=NONE]]]

2016-03-03 22:20:30,194 TRACE [B.defaultRpcServer.handler=20,queue=2,port=16020] hfile.HFileReaderImpl: Cache hit <span class="code-keyword">return</span> [blockType=DATA, fileOffset=2057478, headerSize=33, onDiskSizeWithoutHeader=2024, uncompressedSizeWithoutHeader=2020, prevBlockOffset=2055421, isUseHBaseChecksum=<span class="code-keyword">true</span>, checksumType=CRC32C, bytesPerChecksum=16384, onDiskDataSizeWithHeader=2053, getOnDiskSizeWithHeader=2057, totalChecksumBytes=4, isUnpacked=<span class="code-keyword">true</span>, buf=[org.apache.hadoop.hbase.nio.SingleByteBuff@6063ac95], dataBeginsWith=\x00\x00\x00)\x00\x00\x01`\x00\x16user995698996184959679, fileContext=[usesHBaseChecksum=<span class="code-keyword">true</span>, checksumType=CRC32C, bytesPerChecksum=16384, blocksize=65536, encoding=NONE, includesMvcc=<span class="code-keyword">true</span>, includesTags=<span class="code-keyword">false</span>, compressAlgo=NONE, compressTags=<span class="code-keyword">false</span>, cryptoContext=[cipher=NONE keyHash=NONE]]]

java.lang.Throwable

at org.apache.hadoop.hbase.io.hfile.HFileReaderImpl.readBlock(HFileReaderImpl.java:1515)

at org.apache.hadoop.hbase.io.hfile.HFileReaderImpl$HFileScannerImpl.readNextDataBlock(HFileReaderImpl.java:906)

at org.apache.hadoop.hbase.io.hfile.HFileReaderImpl$HFileScannerImpl.isNextBlock(HFileReaderImpl.java:1106)

at org.apache.hadoop.hbase.io.hfile.HFileReaderImpl$HFileScannerImpl.positionForNextBlock(HFileReaderImpl.java:1100)

at org.apache.hadoop.hbase.io.hfile.HFileReaderImpl$HFileScannerImpl.\_next(HFileReaderImpl.java:1118)

at org.apache.hadoop.hbase.io.hfile.HFileReaderImpl$HFileScannerImpl.next(HFileReaderImpl.java:1139)

at org.apache.hadoop.hbase.regionserver.StoreFileScanner.next(StoreFileScanner.java:152)

at org.apache.hadoop.hbase.regionserver.KeyValueHeap.next(KeyValueHeap.java:114)

at org.apache.hadoop.hbase.regionserver.StoreScanner.next(StoreScanner.java:628)

at org.apache.hadoop.hbase.regionserver.KeyValueHeap.next(KeyValueHeap.java:153)

at org.apache.hadoop.hbase.regionserver.HRegion$RegionScannerImpl.populateResult(HRegion.java:5703)

at org.apache.hadoop.hbase.regionserver.HRegion$RegionScannerImpl.nextInternal(HRegion.java:5849)

at org.apache.hadoop.hbase.regionserver.HRegion$RegionScannerImpl.nextRaw(HRegion.java:5622)

at org.apache.hadoop.hbase.regionserver.HRegion$RegionScannerImpl.next(HRegion.java:5598)

at org.apache.hadoop.hbase.regionserver.HRegion$RegionScannerImpl.next(HRegion.java:5584)

at org.apache.hadoop.hbase.regionserver.RSRpcServices.get(RSRpcServices.java:2187)

at org.apache.hadoop.hbase.regionserver.RSRpcServices.get(RSRpcServices.java:2123)

at org.apache.hadoop.hbase.protobuf.generated.ClientProtos$ClientService$2.callBlockingMethod(ClientProtos.java:33512)

at org.apache.hadoop.hbase.ipc.RpcServer.call(RpcServer.java:2229)

at org.apache.hadoop.hbase.ipc.CallRunner.run(CallRunner.java:109)

at org.apache.hadoop.hbase.ipc.RpcExecutor.consumerLoop(RpcExecutor.java:136)

at org.apache.hadoop.hbase.ipc.RpcExecutor$1.run(RpcExecutor.java:111)

at java.lang.<span class="code-object">Thread</span>.run(<span class="code-object">Thread</span>.java:745)

2016-03-03 22:20:30,195 INFO [B.defaultRpcServer.handler=20,queue=2,port=16020] regionserver.StoreScanner: QCODE DONE

</pre>

</div></div>

<p>We are in StoreScanner#next.</p>

<p>Matcher does SEEK\_NEXT\_COL.<br/>

We pull in the block that has our Cell in it.<br/>

Matcher does INCLUDE<br/>

Then Matcher does SKIP<br/>

SKIP has us go read the next block.<br/>

Then Matcher does DONE.</p>

<p>Why ain't I getting to DONE after INCLUDE?</p>

<p>Any clues?</p>

<p>This code has been like this a while.</p>

Title: [HBASE-15391] Avoid too large "deleted from META" info log

Description: <p>When deleting a large table in HBase, there will be a large info log in HMaster.</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

2016-02-29,05:58:45,920 INFO org.apache.hadoop.hbase.catalog.MetaEditor: Deleted [{ENCODED =&gt; 4b54572150941cd03f5addfdeab0a754, NAME =&gt; 'YCSBTest,,1453186492932.4b54572150941cd03f5addfdeab0a754.', STARTKEY =&gt; '', ENDKEY =&gt; 'user01'}, {ENCODED =&gt; 715e142bcd6a31d7842abf286ef8a5fe, NAME =&gt; 'YCSBTest,user01,1453186492933.715e142bcd6a31d7842abf286ef8a5fe.', STARTKEY =&gt; 'user01', ENDKEY =&gt; 'user02'}, {ENCODED =&gt; 5f9cef5714973f13baa63fba29a68d70, NAME =&gt; 'YCSBTest,user02,1453186492933.5f9cef5714973f13baa63fba29a68d70.', STARTKEY =&gt; 'user02', ENDKEY =&gt; 'user03'}, {ENCODED =&gt; 86cf3fa4c0a6b911275512c1d4b78533, NAME =&gt; 'YCSBTest,user0...

</pre>

</div></div>

<p>The reason is that MetaTableAccessor will log all regions when deleting them from meta. See, MetaTableAccessor.java#deleteRegions</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

<span class="code-keyword">public</span> <span class="code-keyword">static</span> void deleteRegions(Connection connection,

List&lt;HRegionInfo&gt; regionsInfo, <span class="code-object">long</span> ts) <span class="code-keyword">throws</span> IOException {

List&lt;Delete&gt; deletes = <span class="code-keyword">new</span> ArrayList&lt;Delete&gt;(regionsInfo.size());

<span class="code-keyword">for</span> (HRegionInfo hri: regionsInfo) {

Delete e = <span class="code-keyword">new</span> Delete(hri.getRegionName());

e.addFamily(getCatalogFamily(), ts);

deletes.add(e);

}

deleteFromMetaTable(connection, deletes);

LOG.info(<span class="code-quote">"Deleted "</span> + regionsInfo);

}

</pre>

</div></div>

<p>Just change the info log to debug and add a info log about the number of deleted regions. Others suggestions are welcomed~</p>

Title: [HBASE-15390] ClientExceptionUtils doesn't handle CallQueueTooBigException properly

Description: <p>In #isMetaClearingException() we're checking for CallQueueTooBigException, but under debugger in tests I saw that what we're really getting in here is RemoteWithExtrasException, which doesn't allow to easily unwrap CQTBE from it, since it's stored in the classname field, and findException() or unwrapRemoteException() fail to unwrap it correctly.</p>

<p>I think we'd have the same behavior with other exceptions wrapper this way.</p>

Title: [HBASE-15389] Write out multiple files when compaction

Description:

Title: [HBASE-15388] Add ACL for some master methods

Description: <p>Some new methods and some old ones do not have ACLs. </p>

<p>A basic look at the master rpc endpoints results in </p>

<ul class="alternate" type="square">

<li>Catalog janitor methods</li>

<li>set balancer switch</li>

<li>Normalizer methods</li>

<li>split merge switch</li>

<li>mob methods</li>

<li>others?</li>

</ul>

Title: [HBASE-15387] Make HalfStoreFileReader configurable in LoadIncrementalHFiles

Description: <p>Currently we are initializing HalfStoreFileReader to split the HFile but we might have different implementation for splitting. So we can make it configurable. It's needed for local indexing in Phoenix(<a href="https://issues.apache.org/jira/browse/PHOENIX-2736" title="Fix possible data loss with local indexes when there are splits during bulkload" class="issue-link" data-issue-key="PHOENIX-2736">PHOENIX-2736</a>). </p>

Title: [HBASE-15386] PREFETCH\_BLOCKS\_ON\_OPEN in HColumnDescriptor is ignored

Description: <p>We use the global flag hbase.rs.prefetchblocksonopen only and ignore the HCD setting.</p>

<p>Purge from HCD or hook it up again (it probably worked once).</p>

<p>Thanks to Daniel Pol for finding this one. Let me fix.</p>

Title: [HBASE-15385] A failed atomic folder rename operation can never recovery for the destination file is deleted in Wasb filesystem

Description: <p>When using Wsab file system, we found that a failed atomic folder rename operation can never recovery for the destination file deleted in Wasb filesystem. </p>

{quota}

<p>ls: Attempting to complete rename of file hbase/azurtst-xiaomi/data/default/YCSBTest/.tabledesc during folder rename redo, and file was not found in source or destination.</p>

<blockquote></blockquote>

<p>The reason is the the file is renamed to the destination file before the crash, and the destination file is deleted by another process after crash. So the recovery is blocked during finishing the rename operation of this file when found the source and destination files all don't exist.</p>

<p>See: NativeAzureFileSystem.java #finishSingleFileRename</p>

<p>Another serious problem is that the recovery of atomic rename operation may delete new created file which is same name as the source file, because the file system don't check if there are rename operation need be redo.</p>

<p>Suggestions are welcomed~</p>

Title: [HBASE-15384] Avoid using '/tmp' directory in TestBulkLoad

Description:

Title: [HBASE-15383] Load distribute across secondary read replicas for meta

Description: <p>Right now, we always hit the primary replica for meta and fallback to the secondary replicas in case of a timeout. This can hamper performance in scenarios where meta becomes a hot region e.g. cluster ramp up..clients dropping connections etc.</p>

<p>It's good to have a load distribution approach on meta's secondary replicas with fallback to primary if we read stale data.</p>

Title: [HBASE-15382] Expose regionserver metadata (ie groups, tables, servers) via JMX

Description: <p>This feature was removed from the base patch. So we can come up with a proper interface for other components to use as well, as directly accessing jmx is not an option.</p>

Title: [HBASE-15381] Implement a distributed MOB compaction by procedure

Description: <p>In MOB, there is a periodical compaction which runs in HMaster (It can be disabled by configuration), some small mob files are merged into bigger ones. Now the compaction only runs in HMaster which is not efficient and might impact the running of HMaster. In this JIRA, a distributed MOB compaction is introduced, it is triggered by HMaster, but all the compaction jobs are distributed to HRegionServers.</p>

Title: [HBASE-15380] Purge rollback support in Store etc.

Description: <p>Rollback is no longer needed after "<a href="https://issues.apache.org/jira/browse/HBASE-15158" title="Change order in which we do write pipeline operations; do all under row locks!" class="issue-link" data-issue-key="HBASE-15158"><del>HBASE-15158</del></a> Change order in which we do write pipeline operations; do all under row locks". Purge support. Will simplify this segment work.</p>

Title: [HBASE-15379] Fake cells created in read path not implementing SettableSequenceId

Description: <p>This issue found by <a href="https://issues.apache.org/jira/secure/ViewProfile.jspa?name=appy" class="user-hover" rel="appy">Appy</a>. In <a href="https://issues.apache.org/jira/browse/HBASE-14099" title="StoreFile.passesKeyRangeFilter need not create Cells from the Scan&#39;s start and stop Row" class="issue-link" data-issue-key="HBASE-14099"><del>HBASE-14099</del></a> he says,<br/>

I was doing some testing when I hit a weird issue, seems related to this, so re-opening it (apologies in advance if it's not). Here's the stack trace</p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>java.io.IOException: java.lang.UnsupportedOperationException: Cell is not of type org.apache.hadoop.hbase.SettableSequenceId

at org.apache.hadoop.hbase.CellUtil.setSequenceId(CellUtil.java:923)

at org.apache.hadoop.hbase.regionserver.StoreFileScanner.setCurrentCell(StoreFileScanner.java:231)

at org.apache.hadoop.hbase.regionserver.StoreFileScanner.requestSeek(StoreFileScanner.java:389)

at org.apache.hadoop.hbase.regionserver.StoreScanner.seekScanners(StoreScanner.java:348)

at org.apache.hadoop.hbase.regionserver.StoreScanner.&lt;init&gt;(StoreScanner.java:212)

at org.apache.hadoop.hbase.regionserver.HStore.createScanner(HStore.java:1873)

at org.apache.hadoop.hbase.regionserver.HStore.getScanner(HStore.java:1863)

at org.apache.hadoop.hbase.regionserver.HRegion$RegionScannerImpl.&lt;init&gt;(HRegion.java:5487)

at org.apache.hadoop.hbase.regionserver.HRegion.instantiateRegionScanner(HRegion.java:2577)

at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:2563)

at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:2544)

at org.apache.hadoop.hbase.regionserver.HRegion.getScanner(HRegion.java:2534)

at org.apache.hadoop.hbase.regionserver.HRegion.get(HRegion.java:6659)

at org.apache.hadoop.hbase.regionserver.HRegion.get(HRegion.java:6624)

at org.apache.hadoop.hbase.regionserver.TestWithSingleHRegion.test(TestWithSingleHRegion.java:48)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:606)

at org.junit.runners.model.FrameworkMethod$1.runReflectiveCall(FrameworkMethod.java:50)

at org.junit.internal.runners.model.ReflectiveCallable.run(ReflectiveCallable.java:12)

at org.junit.runners.model.FrameworkMethod.invokeExplosively(FrameworkMethod.java:47)

at org.junit.internal.runners.statements.InvokeMethod.evaluate(InvokeMethod.java:17)

at org.junit.runners.ParentRunner.runLeaf(ParentRunner.java:325)

at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:78)

at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:57)

at org.junit.runners.ParentRunner$3.run(ParentRunner.java:290)

at org.junit.runners.ParentRunner$1.schedule(ParentRunner.java:71)

at org.junit.runners.ParentRunner.runChildren(ParentRunner.java:288)

at org.junit.runners.ParentRunner.access$000(ParentRunner.java:58)

at org.junit.runners.ParentRunner$2.evaluate(ParentRunner.java:268)

at org.junit.runners.ParentRunner.run(ParentRunner.java:363)

at org.junit.runner.JUnitCore.run(JUnitCore.java:137)

at com.intellij.junit4.JUnit4IdeaTestRunner.startRunnerWithArgs(JUnit4IdeaTestRunner.java:117)

at com.intellij.rt.execution.junit.JUnitStarter.prepareStreamsAndStart(JUnitStarter.java:234)

at com.intellij.rt.execution.junit.JUnitStarter.main(JUnitStarter.java:74)

</pre>

</div></div>

<p>I think it's because of using changing from KeyValue to a different sub-class of <tt>Cell}}l which doesn't implement {{SettableSequenceId</tt></p>

<div class="preformatted panel" style="border-width: 1px;"><div class="preformattedContent panelContent">

<pre>- this.startKey = KeyValueUtil.createFirstDeleteFamilyOnRow(scan.getStartRow(),

+ this.startKey = CellUtil.createFirstDeleteFamilyCellOnRow(scan.getStartRow(),

</pre>

</div></div>

<p>To replicate it, download the attached hfiles somewhere, copy the TestWithSingleHRegion class to regionserver tests, change the ROOT\_DIR appropriately and run it.</p>

Title: [HBASE-15378] Scanner can not handle a heartbeat message with no results

Description: <p>When a RS scanner get a TIME\_LIMIT\_REACHED\_MID\_ROW state, they will stop scanning and send back what it has read to client and mark the message as a heartbeat message. If there is no cell has been read, it will be an empty response. </p>

<p>However, ClientScanner only handles the situation that the client gets an empty heartbeat and its cache is not empty. If the cache is empty too, it will be regarded as end-of-region and open a new scanner for next region.</p>

Title: [HBASE-15377] Per-RS Get metric is time based, per-region metric is size-based

Description: <p>We have metrics for Get operations at the region server level and region level. </p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

<span class="code-quote">"Get\_num\_ops"</span> : 4837505,

<span class="code-quote">"Get\_min"</span> : 0,

<span class="code-quote">"Get\_max"</span> : 296,

<span class="code-quote">"Get\_mean"</span> : 0.2934618155433431,

<span class="code-quote">"Get\_median"</span> : 0.0,

<span class="code-quote">"Get\_75th\_percentile"</span> : 0.0,

<span class="code-quote">"Get\_95th\_percentile"</span> : 1.0,

<span class="code-quote">"Get\_99th\_percentile"</span> : 1.0,

</pre>

</div></div>

<p>and </p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

<span class="code-quote">"Namespace\_hbase\_table\_meta\_region\_1588230740\_metric\_get\_num\_ops"</span> : 103,

<span class="code-quote">"Namespace\_hbase\_table\_meta\_region\_1588230740\_metric\_get\_min"</span> : 450,

<span class="code-quote">"Namespace\_hbase\_table\_meta\_region\_1588230740\_metric\_get\_max"</span> : 470,

<span class="code-quote">"Namespace\_hbase\_table\_meta\_region\_1588230740\_metric\_get\_mean"</span> : 450.19417475728153,

<span class="code-quote">"Namespace\_hbase\_table\_meta\_region\_1588230740\_metric\_get\_median"</span> : 460.0,

<span class="code-quote">"Namespace\_hbase\_table\_meta\_region\_1588230740\_metric\_get\_75th\_percentile"</span> : 470.0,

<span class="code-quote">"Namespace\_hbase\_table\_meta\_region\_1588230740\_metric\_get\_95th\_percentile"</span> : 470.0,

<span class="code-quote">"Namespace\_hbase\_table\_meta\_region\_1588230740\_metric\_get\_99th\_percentile"</span> : 470.0,

</pre>

</div></div>

<p>The problem is that the report values for the region server shows the latency, versus the reported values for the region shows the response sizes. There is no way of telling this without reading the source code. </p>

<p>I think we should deprecate response size histograms in favor of latency histograms. </p>

<p>See also <a href="https://issues.apache.org/jira/browse/HBASE-15376" title="ScanNext metric is size-based while every other per-operation metric is time based" class="issue-link" data-issue-key="HBASE-15376"><del>HBASE-15376</del></a>. </p>

Title: [HBASE-15376] ScanNext metric is size-based while every other per-operation metric is time based

Description: <p>We have per-operation metrics for <tt>Get</tt>, <tt>Mutate</tt>, <tt>Delete</tt>, <tt>Increment</tt>, and <tt>ScanNext</tt>. </p>

<p>The metrics are emitted like: </p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

<span class="code-quote">"Get\_num\_ops"</span> : 4837505,

<span class="code-quote">"Get\_min"</span> : 0,

<span class="code-quote">"Get\_max"</span> : 296,

<span class="code-quote">"Get\_mean"</span> : 0.2934618155433431,

<span class="code-quote">"Get\_median"</span> : 0.0,

<span class="code-quote">"Get\_75th\_percentile"</span> : 0.0,

<span class="code-quote">"Get\_95th\_percentile"</span> : 1.0,

<span class="code-quote">"Get\_99th\_percentile"</span> : 1.0,

...

<span class="code-quote">"ScanNext\_num\_ops"</span> : 194705,

<span class="code-quote">"ScanNext\_min"</span> : 0,

<span class="code-quote">"ScanNext\_max"</span> : 18441,

<span class="code-quote">"ScanNext\_mean"</span> : 7468.274651395701,

<span class="code-quote">"ScanNext\_median"</span> : 583.0,

<span class="code-quote">"ScanNext\_75th\_percentile"</span> : 583.0,

<span class="code-quote">"ScanNext\_95th\_percentile"</span> : 13481.0,

<span class="code-quote">"ScanNext\_99th\_percentile"</span> : 13481.0,

</pre>

</div></div>

<p>The problem is that all of Get,Mutate,Delete,Increment,Append,Replay are time based tracking how long the operation ran, while ScanNext is tracking returned response sizes (returned cell-sizes to be exact). Obviously, this is very confusing and you would only know this subtlety if you read the metrics collection code. </p>

<p>Not sure how useful is the ScanNext metric as it is today. We can deprecate it, and introduce a time based one to keep track of scan request latencies. </p>

<p>ps. Shamelessly using the parent jira (since these seem relavant). </p>

Title: [HBASE-15375] Do not write to '/tmp' in TestRegionMover

Description:

Title: [HBASE-15374] Avoid using '/tmp' directory in our unit tests

Description: <p>I'm used to run unit tests on a remote server before summit a large patch because sometimes a testcase failure can not be reproduced locally.</p>

<p>Recently <a href="https://issues.apache.org/jira/secure/ViewProfile.jspa?name=yangzhe1991" class="user-hover" rel="yangzhe1991">Phil Yang</a> and me share a remote server with a different account. We found that some of the unit tests write to '/tmp' with a fixed name and do not clean the file after test. This cause that the unit test can only be succeeded for one person...</p>

<p>Open an umbrella issue to address this.</p>

Title: [HBASE-15373] DEPRECATED\_NAME\_OF\_NO\_LIMIT\_THROUGHPUT\_CONTROLLER\_CLASS value is wrong in CompactionThroughputControllerFactory

Description: <p>I couldn't turn off compaction throughput by following release note instructions. I fixed release notes in parent but also needs code fix in factory class.</p>

Title: [HBASE-15372] HBase Backup/Restore Phase 2: Keep WAL files in a centralized place to avoid duplication

Description: <p>Currently, we keep WAL file per table in backup site, this creates significant data duplication in case of many tables in a backup set. We have to keep all WAL files in a single place and keep track of WAL files involved per table/backup id as links (references) </p>

<p>This is not only the data duplication issue, but a performance issue as well (we copy the same file over and over again for every table in a backup set). </p>

Title: [HBASE-15371] Procedure V2 - Completed support parent-child procedure

Description: <p>In Procedure-V2 Phase 1 (<a href="https://issues.apache.org/jira/browse/HBASE-14336" title="Procedure V2 Phase 1 - Procedure Framework and Making DDL Operations fault tolerant" class="issue-link" data-issue-key="HBASE-14336"><del>HBASE-14336</del></a>), some infrastructure of supporting child procedure exists. However, there is no need in Phase 1 (master DDL) to have multi-level procedures. This JIRA implements adding child procedures to procedure execution list and execute them before parent procedure can make further progress.</p>

Title: [HBASE-15370] Backport Moderate Object Storage (MOB) to branch-1

Description: <p>MOB feature was integrated to master branch half a year ago.</p>

<p>Since then there have been bug fixes which stabilize the feature.<br/>

Some customers have been using it at PB scale.</p>

<p>Here is discussion thread on dev mailing list:</p>

<p><a href="http://search-hadoop.com/m/YGbbDSqxD1PYXK62/hbase+MOB+in+branch-1&amp;subj=Re+MOB+in+branch+1+Re+RESULT+VOTE+Merge+branch+hbase+11339+HBase+MOB+to+trunk+" class="external-link" rel="nofollow">http://search-hadoop.com/m/YGbbDSqxD1PYXK62/hbase+MOB+in+branch-1&amp;subj=Re+MOB+in+branch+1+Re+RESULT+VOTE+Merge+branch+hbase+11339+HBase+MOB+to+trunk+</a></p>

<p>This issue is to backport MOB feature to branch-1.</p>

Title: [HBASE-15369] Handle NPE in region.jsp

Description: <p>If empty or non-existing region name parameter is passed to "region.jsp ?name=" it will cause 500 NPE error. It is not big deal but looks ugly.</p>

Title: [HBASE-15368] Add relative window support

Description: <p>To better determine 'hot' data.</p>

Title: [HBASE-15367] Configuring DisabledWAL provider hangs the RegionServer

Description: <p>DisabledWALProvider is not for production cases. But still configuring one does not allow region opening to work fine and thus hanging create table process.<br/>

[code}<br/>

at org.apache.hadoop.hbase.wal.WALKey.getWriteEntry(WALKey.java:106)<br/>

at org.apache.hadoop.hbase.regionserver.wal.WALUtil.writeMarker(WALUtil.java:133)<br/>

at org.apache.hadoop.hbase.regionserver.wal.WALUtil.writeRegionEventMarker(WALUtil.java:88)<br/>

at org.apache.hadoop.hbase.regionserver.HRegion.writeRegionOpenMarker(HRegion.java:1006)<br/>

at org.apache.hadoop.hbase.regionserver.HRegion.openHRegion(HRegion.java:6483)<br/>

at org.apache.hadoop.hbase.regionserver.HRegion.openHRegion(HRegion.java:6433)<br/>

at org.apache.hadoop.hbase.regionserver.HRegion.openHRegion(HRegion.java:6404)<br/>

at org.apache.hadoop.hbase.regionserver.HRegion.openHRegion(HRegion.java:6360)<br/>

at org.apache.hadoop.hbase.regionserver.HRegion.openHRegion(HRegion.java:6311)<br/>

at org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler.openRegion(OpenRegionHandler.java:282)<br/>

at org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler.process(OpenRegionHandler.java:107)<br/>

at org.apache.hadoop.hbase.executor.EventHandler.run(EventHandler.java:104)<br/>

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)<br/>

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

But there is also a clear comment in WALUtil#writeMarker

</pre>

</div></div>

<p> // If you get hung here, is it a real WAL or a mocked WAL? If the latter, you need to<br/>

// trip the latch that is inside in getWriteEntry up in your mock. See down in the append<br/>

// called from onEvent in FSHLog.</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java"></pre>

</div></div>

<p>But this is about Mocking but here it is not mocking and still we are not able to use it. The only option is like the client should say SKIP\_WAL for every mutation if we really need to avoid WAL.</p>

Title: [HBASE-15366] Add doc, trace-level logging, and test around hfileblock

Description: <p>What hfileblock is doing &#8211; that it overreads when pulling in from hdfs to fetch the header of the next block to save on seeks; that it caches the block and overread and then adds an extra 13 bytes to the cached entry; that buckets in bucketcache have at least four hfileblocks in them and so on &#8211; was totally baffling me. This patch docs the class, adds some trace-level logging so you can see if you are doing the right thing, and then adds a test of file-backed bucketcache that checks that persistence is working.</p>

Title: [HBASE-15365] Do not write to '/tmp' in TestHBaseConfiguration

Description: <p>In testGetPassword, we create a key file at /tmp/foo.jks and set its permissions to 600. This will cause testcase failure on a shared machine.</p>

Title: [HBASE-15364] Fix unescaped < characters in Javadoc

Description: <p>From <a href="https://builds.apache.org/job/HBase%20Website%20Link%20Ckecker/28/artifact/link\_report/index.html:" class="external-link" rel="nofollow">https://builds.apache.org/job/HBase%20Website%20Link%20Ckecker/28/artifact/link\_report/index.html:</a> </p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

host: hbase.apache.org

date: Mon, 29-Feb-2016 12:06:21 (local)

Linklint version: 2.3.5\_ingo\_020

#------------------------------------------------------------

# warn 2 warnings (cross referenced)

#------------------------------------------------------------

unquoted <span class="code-quote">"&lt;"</span> in &lt;0.90.5, &lt;0.90.5, &lt;

occurred in

/devapidocs/org/apache/hadoop/hbase/util/HBaseFsck.html

unquoted <span class="code-quote">"&lt;"</span> in &lt;0.92.0) a master

res

occurred in

/devapidocs/org/apache/hadoop/hbase/util/HBaseFsck.html

</pre>

</div></div>

Title: [HBASE-15363] Add client side metrics for SASL connection failures

Description: <p>There are a number of cases where we can get SASL connection failures before getting to the server, like errors talking to the KDC/TGS and misconfiguration of kerberos principals. Hence these will not show up in the server-side authentication\_failures metric.</p>

<p>We should add client side metrics on SASL connection failures to capture these.</p>

Title: [HBASE-15362] Compression Algorithm does not respect config params from hbase-site

Description: <p>Compression creates conf using new Configuration() and this leads to it not respecting the confs set in hbase-site, fixing it is trivial using HBaseConfiguration.create()</p>

Title: [HBASE-15361] Remove unnecessary or Document constraints on BucketCache possible bucket sizes

Description: <p>When we were trying to tune the bucket sizes <tt>hbase.bucketcache.bucket.sizes</tt> according to our workload, we encountered an issue due to the way offset is stored in the bucket entry. We divide the offset into integer base and byte value and it assumes that all bucket offsets will be a multiple of 256 (left shifting by 8). See the code below</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

<span class="code-object">long</span> offset() { <span class="code-comment">// Java has no unsigned numbers

</span> <span class="code-object">long</span> o = ((<span class="code-object">long</span>) offsetBase) &amp; 0xFFFFFFFF;

o += (((<span class="code-object">long</span>) (offset1)) &amp; 0xFF) &lt;&lt; 32;

<span class="code-keyword">return</span> o &lt;&lt; 8;

}

<span class="code-keyword">private</span> void setOffset(<span class="code-object">long</span> value) {

<span class="code-keyword">assert</span> (value &amp; 0xFF) == 0;

value &gt;&gt;= 8;

offsetBase = (<span class="code-object">int</span>) value;

offset1 = (<span class="code-object">byte</span>) (value &gt;&gt; 32);

}

</pre>

</div></div>

<p>This was there to save 3 bytes per BucketEntry instead of using long and when there are no other fields in the Bucket Entry, but now there are lot of fields in the bucket entry , This not documented so we could either document the constraint that it should be a strict 256 bytes multiple of just go away with this constraint. </p>

Title: [HBASE-15360] Fix flaky TestSimpleRpcScheduler

Description: <p>There were several flaky tests added there as part of <a href="https://issues.apache.org/jira/browse/HBASE-15306" title="Make RPC call queue length dynamically configurable" class="issue-link" data-issue-key="HBASE-15306"><del>HBASE-15306</del></a> and likely <a href="https://issues.apache.org/jira/browse/HBASE-15136" title="Explore different queuing behaviors while busy" class="issue-link" data-issue-key="HBASE-15136"><del>HBASE-15136</del></a>.</p>

Title: [HBASE-15359] Simplifying Segment hierarchy

Description: <p>Now that it is clear that no memstore segment will be implemented as an HFIle, and that all segments store their data in some representation of CellSet (skip-list or flat), the segment hierarchy can be much simplified.</p>

<p>The attached patch includes only 3 classes in the hierarchy:<br/>

Segment - comprises most of the state and implementation<br/>

MutableSegment - extends API with add and rollback functionality<br/>

ImmutableSegment - extends API with key-value scanner for snapshot</p>

<p>SegmentScanner is the scanner for all types of segments. </p>

<p>In addition, the option to rollback immutable segment in the memstore is disabled.</p>

<p>This code would allow us to make progress independently in the compaction subtask (<a href="https://issues.apache.org/jira/browse/HBASE-14920" title="Compacting Memstore" class="issue-link" data-issue-key="HBASE-14920">HBASE-14920</a>) and the flat index representation subtask (<a href="https://issues.apache.org/jira/browse/HBASE-14921" title="Memory optimizations" class="issue-link" data-issue-key="HBASE-14921">HBASE-14921</a>). It also means that the new immutable segment can reuse the existing SegmentScanner, instead of implementing a new scanner.</p>

Title: [HBASE-15358] canEnforceTimeLimitFromScope should use timeScope instead of sizeScope

Description: <p>A small but maybe critical bug</p>

Title: [HBASE-15357] TableInputFormatBase getSplitKey does not handle signed bytes correctly

Description: <p>When auto-balance is enabled in TableInputFormatBase and the table key is a binary key, the getSplitKey method does not function correctly for signed bytes. The proposed solution it to utilize org.apache.hadoop.hbase.util.Bytes#split method to find the split key. org.apache.hadoop.hbase.util.Bytes#split is stated to be a expensive operation, so if another solution is preferred, that is fine. In addition, handling of a split key that is equal to the TableSplit end key is added to calculateRebalancedSplits.</p>

Title: [HBASE-15356] Remove unused Imports

Description: <p>Remove unused Imports.</p>

Title: [HBASE-15355] region.jsp can not be found on info server of master

Description: <p>After <a href="https://issues.apache.org/jira/browse/HBASE-10569" class="external-link" rel="nofollow">HBASE-10569</a>, master is also a regionserver and it will serve regions of system tables. The meta region info could be viewed on master at the address such as : <a href="http://localhost:16010/region.jsp?name=1588230740" class="external-link" rel="nofollow">http://localhost:16010/region.jsp?name=1588230740</a>. The real path of region.jsp for the request will be hbase-webapps/master/region.jsp on master, however, the region.jsp is under the directory hbase-webapps/regionserver, so that can not be found on master.</p>

Title: [HBASE-15354] Use same criteria for clearing meta cache for all operations

Description: <p>Currently we do not clear/update meta cache for some special exceptions if the operation went through AsyncProcess#submit like HTable#put calls. But, we clear meta cache without checking for these special exceptions in case of other operations like gets, deletes etc because they directly go through the RpcRetryingCaller#callWithRetries instead of the AsyncProcess. </p>

Title: [HBASE-15353] Add metric for number of CallQueueTooBigException's

Description: <p>This exception is being thrown more. We should add a metric for this one.</p>

Title: [HBASE-15352] FST BlockEncoder

Description: <p>We could improve on the existing <a href="http://hbase.apache.org/devapidocs/org/apache/hadoop/hbase/codec/prefixtree/package-summary.html" class="external-link" rel="nofollow">PREFIX\_TREE block</a> encoder by upgrading the persistent data structure from a trie to a finite state transducer. This would theoretically allow us to reuse bytes not just for rowkey prefixes, but infixes and suffixes as well. My read of the literature means we may also be able to encode values as well, further reducing storage size when values are repeated (ie, a "customer id" field with very low cardinality &#8211; probably happens a lot in our denormalized world). There's a really nice <a href="http://blog.burntsushi.net/transducers/" class="external-link" rel="nofollow">blog post</a> about this data structure, and apparently our siblings in Lucene make heavy use of <a href="http://lucene.apache.org/core/5\_5\_0/core/org/apache/lucene/util/fst/package-summary.html#package\_description" class="external-link" rel="nofollow">their implementation</a>.</p>

Title: [HBASE-15351] Fix description of hbase.bucketcache.size in hbase-default.xml

Description: <p>As suggested in the parent issue, a fix in the doc needs to be repeated in hbase-default.xml</p>

Title: [HBASE-15350] Enable individual unit test in hbase-spark module

Description:

Title: [HBASE-15349] Update surefire version to 2.19.1

Description: <p>So that new properties like surefire.excludesFile and includesFile can be used to easily exclude/include flaky tests.</p>

Title: [HBASE-15348] Fix tests broken by recent metrics re-work

Description: <p>Counts are appoximate and go away. We should re-work the tests or test utils to make them work now.</p>

Title: [HBASE-15347] Update CHANGES.txt for 1.3

Description:

Title: [HBASE-15346] add 1.3 RM to docs

Description:

Title: [HBASE-15345] add branch-1.3 post-commit builds

Description: <p>point hbase-1.3 build to branch-1.3, add hbase-1.4 and point it to branch-1.</p>

Title: [HBASE-15344] add 1.3 to prereq tables in ref guide

Description:

Title: [HBASE-15343] add branch-1.3 to precommit branches

Description:

Title: [HBASE-15342] create branch-1.3 and update branch-1 poms to 1.4.0-SNAPSHOT

Description: <p>create branch-1.3 and update branch-1 poms to 1.4.0-SNAPSHOT</p>

Title: [HBASE-15341] 1.3 release umbrella

Description: <p>Umbrella jira for 1.3 release.</p>

Title: [HBASE-15340] Partial row result of scan may return data violates the row-level transaction

Description: <p>There are cases the region sever will return partial row result, such as the client set batch for scan or configured size limit reached. In these situations, the client may return data that violates the row-level transaction to the application. The following steps show the problem:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

<span class="code-comment">// assume there is a test table 'test\_table' with one family 'F' and one region 'region'.

</span><span class="code-comment">// meanwhile there are two region servers 'rsA' and 'rsB'.

</span>1. Let 'region' firstly located in 'rsA' and put one row with two columns 'c1' and 'c2' as:

&gt; put 'test\_table', 'row', 'F:c1', 'value1', 'F:c2', 'value1'

2. Start a client to scan 'test\_table', with scan.setBatch(1) and scan.setCaching(1). The client will get one column as : {column='F:c1' and value='value1'} in the first rpc call after scanner created, and the result will be returned to application.

3. Before the client issues the next request, the 'region' was moved to 'rsB' and accepted another mutations <span class="code-keyword">for</span> the two columns 'c1' and 'c2' as:

&gt; put 'test\_table', 'row', 'F:c1', 'value2', 'F:c2', 'value2'

4. Then, the client will receive a RegionMovedException when issuing next request and will retry to open scanner on 'rsB'. The newly opened scanner will higher mvcc than old data so that could read out column as : { column='F:c2' with value='value2'} and <span class="code-keyword">return</span> the result to application.

Therefore, the application will get data as:

'row' column='F:c1' value='value1'

'row' column='F:c2', value='value2'

The returned data is combined from two different mutations and violates the row-level transaction.

</pre>

</div></div>

<p>The reason is that the newly opened scanner after region moved will get a different mvcc. I am not sure whether this result is by design for scan if partial row result is allowed. However, such row result combined from different transactions may make the application have unexpected state.</p>

Title: [HBASE-15339] Improve DateTieredCompactionPolicy

Description: <p>For our MiCloud service, the old data is rarely touched but we still need to keep it, so we want to put the data on inexpensive device and reduce redundancy using EC to cut down the cost.</p>

<p>With date based tiered compaction introduced in <a href="https://issues.apache.org/jira/browse/HBASE-15181" title="A simple implementation of date based tiered compaction" class="issue-link" data-issue-key="HBASE-15181"><del>HBASE-15181</del></a>, new data and old data can be placed in different tier. But the tier boundary moves as time lapse so it is still possible that we do compaction on old tier which breaks our block moving and EC work.</p>

<p>So here we want to introduce an "archive tier" to better fit our scenario. Add an configuration called "archive unit", for example, year. That means, if we find that the tier boundary is already in the previous year, then we reset the boundary to the start of year and end of year, and if we want to do compaction in this tier, just compact all files into one file. The file will never be changed unless we force a major compaction so it is safe to apply EC and other cost reducing approach on the file. And we make more tiers before this tier year by year. </p>

Title: [HBASE-15338] Add a option to disable the data block cache for testing the performance of underlying file system

Description: <p>When testing and comparing the performance of different file systems(HDFS, Azure blob storage, AWS S3 and so on) for HBase, it's better to avoid the affect of the HBase BlockCache and get the actually random read latency when data block is read from underlying file system. (Usually, the index block and meta block should be cached in memory in the testing).</p>

<p>So we add a option in CacheConfig to disable the data block cache.</p>

<p>Suggestions are welcomed~ Thanks</p>

Title: [HBASE-15337] Document FIFO and date tiered compaction in the book

Description: <p>We have two new compaction algorithms FIFO and Date tiered that are for time series data. We should document how to use them in the book. </p>

Title: [HBASE-15336] Support Dataframe writer to the connector

Description: <p>Currently, the connector only support read path. A complete solution should support both read and writer. This subtask add write support.</p>

Title: [HBASE-15335] Add composite key support in row key

Description: <p>Add composite key filter support in the connector.</p>

Title: [HBASE-15334] Add avro support for spark hbase connector

Description: <p>Avro is a popular format for hbase storage. User may want the support natively in the connector.</p>

Title: [HBASE-15333] Enhance the filter to handle short, integer, long, float and double

Description: <p>Currently, the range filter is based on the order of bytes. But for java primitive type, such as short, int, long, double, float, etc, their order is not consistent with their byte order, extra manipulation has to be in place to take care of them correctly.</p>

<p>For example, for the integer range (-100, 100), the filter &lt;= 1, the current filter will return 0 and 1, and the right return value should be (-100, 1]</p>

Title: [HBASE-15332] Document how to take advantage of HDFS-6133 in HBase

Description:

Title: [HBASE-15331] HBase Backup/Restore Phase 2: Optimized Restore operation

Description: <p>The current implementation for restore uses WALReplay M/R job. This has performance and stability problems, since it uses HBase client API to insert data. We have to migrate to bulk load approach: generate hfiles directly from snapshot and incremental images. We run separate M/R job for every backup image between last FULL backup and current incremental backup we restore to and for every table in the list (image). If we have 10 tables and 30 days of incremental backup images - this results in 30x10 = 300 M/R jobs. MUST be optimized.</p>

Title: [HBASE-15330] HBase Backup/Restore Phase 3: support delete/truncate table

Description: <p>Currently, we do not track delete/recreate or truncate table events</p>

Title: [HBASE-15329] Cross-Site Scripting: Reflected in table.jsp

Description: <p>Minor issue where we write back table name in a few places. Should clean it up:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

} <span class="code-keyword">else</span> {

out.write(<span class="code-quote">"\n &lt;title&gt;Table: "</span>);

out.print( fqtn );

out.write(<span class="code-quote">"&lt;/title&gt;\n "</span>);

}

</pre>

</div></div>

Title: [HBASE-15328] Unvalidated Redirect in HMaster

Description: <p>See OWASP page on why we should clean it up someday:</p>

<p><a href="https://www.owasp.org/index.php/Unvalidated\_Redirects\_and\_Forwards\_Cheat\_Sheet" class="external-link" rel="nofollow">https://www.owasp.org/index.php/Unvalidated\_Redirects\_and\_Forwards\_Cheat\_Sheet</a></p>

<p>Here is where we do the redirect:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

@Override

<span class="code-keyword">public</span> void doGet(HttpServletRequest request,

HttpServletResponse response) <span class="code-keyword">throws</span> ServletException, IOException {

<span class="code-object">String</span> redirectUrl = request.getScheme() + <span class="code-quote">":<span class="code-comment">//"</span>

</span> + request.getServerName() + <span class="code-quote">":"</span> + regionServerInfoPort

+ request.getRequestURI();

response.sendRedirect(redirectUrl);

}

}

</pre>

</div></div>

Title: [HBASE-15327] Canary will always invoke admin.balancer() in each sniffing period when writeSniffing is enabled

Description: <p>When Canary#writeSniffing is enabled, Canary#checkWriteTableDistribution will make sure the regions of write table distributed on all region servers as:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

<span class="code-object">int</span> numberOfServers = admin.getClusterStatus().getServers().size();

......

<span class="code-object">int</span> numberOfCoveredServers = serverSet.size();

<span class="code-keyword">if</span> (numberOfCoveredServers &lt; numberOfServers) {

admin.balancer();

}

</pre>

</div></div>

<p>The master will also work as a regionserver, so that ClusterStatus#getServers will contain the master. On the other hand, write table of Canary will not be assigned to master, making numberOfCoveredServers always smaller than numberOfServers and admin.balancer always be invoked in each sniffing period. This may cause frequent region moves. A simple fix is excluding master from numberOfServers.</p>

Title: [HBASE-15326] NPE in TestHRegion.testBatchPut\_whileNoRowLocksHeld

Description: <div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

java.lang.NullPointerException

at org.apache.hadoop.metrics2.lib.MutableHistogram.updateSnapshotMetrics(MutableHistogram.java:72)

at org.apache.hadoop.metrics2.lib.MutableRangeHistogram.snapshot(MutableRangeHistogram.java:59)

at org.apache.hadoop.metrics2.lib.DynamicMetricsRegistry.snapshot(DynamicMetricsRegistry.java:391)

at org.apache.hadoop.hbase.metrics.BaseSourceImpl.getMetrics(BaseSourceImpl.java:146)

at org.apache.hadoop.hbase.test.MetricsAssertHelperImpl.getMetrics(MetricsAssertHelperImpl.java:243)

at org.apache.hadoop.hbase.test.MetricsAssertHelperImpl.getCounter(MetricsAssertHelperImpl.java:201)

at org.apache.hadoop.hbase.regionserver.TestHRegion.testBatchPut\_whileNoRowLocksHeld(TestHRegion.java:1498)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:497)

at org.junit.runners.model.FrameworkMethod$1.runReflectiveCall(FrameworkMethod.java:50)

at org.junit.internal.runners.model.ReflectiveCallable.run(ReflectiveCallable.java:12)

at org.junit.runners.model.FrameworkMethod.invokeExplosively(FrameworkMethod.java:47)

at org.junit.internal.runners.statements.InvokeMethod.evaluate(InvokeMethod.java:17)

at org.junit.internal.runners.statements.RunBefores.evaluate(RunBefores.java:26)

at org.junit.internal.runners.statements.RunAfters.evaluate(RunAfters.java:27)

at org.junit.rules.TestWatcher$1.evaluate(TestWatcher.java:55)

at org.junit.rules.RunRules.evaluate(RunRules.java:20)

at org.junit.runners.ParentRunner.runLeaf(ParentRunner.java:325)

at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:78)

at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:57)

at org.junit.runners.ParentRunner$3.run(ParentRunner.java:290)

at org.junit.runners.ParentRunner$1.schedule(ParentRunner.java:71)

at org.junit.runners.ParentRunner.runChildren(ParentRunner.java:288)

at org.junit.runners.ParentRunner.access$000(ParentRunner.java:58)

at org.junit.runners.ParentRunner$2.evaluate(ParentRunner.java:268)

at org.junit.runners.ParentRunner.run(ParentRunner.java:363)

at org.junit.runner.JUnitCore.run(JUnitCore.java:137)

at com.intellij.junit4.JUnit4IdeaTestRunner.startRunnerWithArgs(JUnit4IdeaTestRunner.java:117)

at com.intellij.rt.execution.junit.JUnitStarter.prepareStreamsAndStart(JUnitStarter.java:234)

at com.intellij.rt.execution.junit.JUnitStarter.main(JUnitStarter.java:74)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:497)

at com.intellij.rt.execution.application.AppMain.main(AppMain.java:144)

</pre>

</div></div>

<p>It seems to be introduced after <a href="https://issues.apache.org/jira/browse/HBASE-15222" title="Use less contended classes for metrics" class="issue-link" data-issue-key="HBASE-15222"><del>HBASE-15222</del></a>, <a href="https://issues.apache.org/jira/secure/ViewProfile.jspa?name=eclark" class="user-hover" rel="eclark">Elliott Clark</a></p>

Title: [HBASE-15325] ResultScanner allowing partial result will miss the rest of the row if the region is moved between two rpc requests

Description: <p><a href="https://issues.apache.org/jira/browse/HBASE-11544" title="[Ergonomics] hbase.client.scanner.caching is dogged and will try to return batch even if it means OOME" class="issue-link" data-issue-key="HBASE-11544"><del>HBASE-11544</del></a> allow scan rpc return partial of a row to reduce memory usage for one rpc request. And client can setAllowPartial or setBatch to get several cells in a row instead of the whole row.</p>

<p>However, the status of the scanner is saved on server and we need this to get the next part if there is a partial result before. If we move the region to another RS, client will get a NotServingRegionException and open a new scanner to the new RS which will be regarded as a new scan from the end of this row. So the rest cells of the row of last result will be missing.</p>

Title: [HBASE-15324] Jitter may cause desiredMaxFileSize overflow in ConstantSizeRegionSplitPolicy and trigger unexpected split

Description: <p>We introduce jitter for region split decision in <a href="https://issues.apache.org/jira/browse/HBASE-13412" title="Region split decisions should have jitter" class="issue-link" data-issue-key="HBASE-13412"><del>HBASE-13412</del></a>, but the following line in <tt>ConstantSizeRegionSplitPolicy</tt> may cause long value overflow if MAX\_FILESIZE is specified to Long.MAX\_VALUE:</p>

<div class="code panel" style="border-width: 1px;"><div class="codeContent panelContent">

<pre class="code-java">

<span class="code-keyword">this</span>.desiredMaxFileSize += (<span class="code-object">long</span>)(desiredMaxFileSize \* (RANDOM.nextFloat() - 0.5D) \* jitter);

</pre>

</div></div>

<p>In our case we specify MAX\_FILESIZE to Long.MAX\_VALUE to prevent target region to split.</p>

Title: [HBASE-15323] Hbase Rest CheckAndDeleteAPi should be able to delete more cells

Description: <p>Java CheckAndDelete API accepts Delete object which can be used to delete (a cell / cell version / multiple cells / column family or a row), but the rest api only allows to delete the cell (without any version)</p>

<p>Need to add this capability to rest api.</p>