Item 12
git_comments:

1. prefix bootstrapper.jar
2. converts C char* to Java byte[]
3. * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
4. return the Admin instance for the local cluster
5. returns the Configuration for the cluster
6. this should be converted to a globalref I think to avoid the underlying java obj getting GC'ed
7. moves region to server
8. construct a new FileDescriptor
9. namespace hbase
10. * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
11. returns the Configuration instance for the cluster
12. moves region to server
13. Does Put into table for family fam, qualifier col with value
14. returns the value for config key retrieved from cluster
15. the hbase-site.xml would be persisted by MiniCluster
16. * * Starts mini hbase cluster with specified number of region servers

git_commits:

1. **summary:** HBASE-17680 Run mini cluster through JNI in tests
   message: HBASE-17680 Run mini cluster through JNI in tests

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Run mini cluster through JNI in tests
   description: Currently tests start local hbase cluster through hbase shell. There is less control over the configuration of the local cluster this way. This issue would replace hbase shell with JNI interface to mini cluster. We would have full control over the cluster behavior. Thanks to [~devaraj] who started this initiative.

jira_issues_comments:

1. **body:** docker hardcodes CLASSPATH as "/usr/src/buck/build/bootstrapper/bootstrapper.jar" Workaround is to embed the output of "bin/hbase classpath" command in a file (/tmp/clspath) which would be read by MiniCluster#create_vm()
   label: code-design
2. We already write a cache of classpath at ./target/cached_classpath.txt
3. The two tests where shell command is replaced: {code} RESULTS FOR //core:location-cache-test PASS 5.2s 3 Passed 0 Skipped 0 Failed //core:location-cache-test TESTS PASSED RESULTS FOR //core:client-test PASS 2.8s 6 Passed 0 Skipped 0 Failed //core:client-test TESTS PASSED {code}
4. **body:** Patch v3 removes commented out code. TestUtil is moved to test class level since TestUtil ctor starts mini cluster which should be reused by all the subtests in the same test class.
   label: code-design
5. In patch v8: cached_classpath.txt is used to pick up CLASSPATHs formatting changes TestUtil(int) ctor is added which allows specification of number of region servers. replaced printf with LOG statements
6. **body:** Thanks for working on this [~devaraj] and [~ted_yu]. This helps a lot in reducing the test execution time as well, since we do not have to wait for java and jruby instantiation multiple times. If I interpret the above, the run times goes from ~60 secs to <10 secs. Plus, we need the multi-regionserver capabilities as well as killing servers, etc for the native client tests which are already available in the mini hbase cluster. For the patch: - Can we move the mini-cluster to be under {{test-util}} module? It does not belong in core. - I really like the way that we can call any method from HTU, Connection, Admin, Table etc, but most of the code in

mini-cluster.cc is unnecessarily in the cpp side. Can we do this instead. Create a NativeTestingUtility.java in hbase-server, and also maybe merge mini-cluster to test-util.cc. The java counter-part will contain almost all of the code that we need to invoke from native (like writeConf(), create_table(), etc in the mini-cluster. This will be better, because almost all of the code will be in the Java side which is way more maintainable. mini-cluster.cc will just call the corresponding java method in NativeTestingUtility class. - Notice that methods like tablePut, create_table, etc will be dramatically simpler. - Method names in cpp usually use camel case with initial upper case, so methods like {{start_cluster()}} should be named {{StartCluster()}}. - You also need to call {{DestroyJavaVM()}} once the testing is done. Maybe add it to Shutdown() or something. - I think this {{+ compiler_flags = ['-I', '/usr/lib/jvm/java-8-openjdk-amd64/include/', '-I', '/usr/lib/jvm/java-8-openjdk-amd64/include/linux'], }} assumes that we are running in the docker container. Our builds are weird, where there is the buck based build and the make based build. However, both are supposed to work without the docker env. So maybe we need to source the JAVA_HOME.
label: code-design

7. Not sure of why we should create NativeTestingUtility - it would be wrapper around HTU and is written in Java, not C++. When native client calls NativeTestingUtility, same translation needs to be carried out. bq. maybe merge mini-cluster to test-util.cc I can do the above.
8. **body:** Couple of more comments: - At least some of the methods are C-style, and rest are C++ style. We can just stick with C++ and encapsulate everything inside the class. - Also, you can use much nicer ifstream instead of fopen / flose, and use std::string instead of mallocing c-strings. You can use {{string::c_str()}} to pass back to the JVM args. - Instead of {{(*env).Foo()}}, you should use {{env->Foo()}}.
   label: code-design
9. **body:** bq. Not sure of why we should create NativeTestingUtility - it would be wrapper around HTU and is written in Java, not C++. The difference is that you do not need to have to code in the JNI layer and the code will be significantly less, as well as more maintainable. The only thing you need to pass back is a return results. For example, in the tests, usually we create a table with a given family, and put some data. You just need two methods which directly forward the request to the java class in this case, no need to import Put object, Table object, Admin, etc.
   label: code-design
10. w.r.t. compiler_flags, on Mac, there is no $JAVA_HOME/include/linux directory. While on docker VM, $JAVA_HOME/include/linux is needed for including jni.h
11. I think for now it's fine to make the whole thing work via Buck (that assumes docker). For the Makefile based builds we can read JAVA_HOME, etc. It makes sense to write the Java wrapper for the HTU that combines various operations, and write a thinner layer for the JNI.
12. **body:** Performed "mvn install -DskipTests" on docker (running on Mac): {code} [INFO] Apache HBase - Shaded - Server .... SUCCESS [14.793 s] [INFO] Apache HBase - Archetypes .... SUCCESS [0.220 s] [INFO] Apache HBase - Exemplar for hbase-client archetype . SUCCESS [20.397 s] [INFO] Apache HBase - Exemplar for hbase-shaded-client archetype SUCCESS [23.015 s] [INFO] Apache HBase - Archetype builder .... SUCCESS [42.005 s] [INFO] -- [INFO] BUILD SUCCESS [INFO] -- [INFO] Total time: 16:33 min [INFO] Finished at: 2017-02-24T16:33:22+00:00 {code} On Mac, same command took: {code} [INFO] -- [INFO] BUILD SUCCESS [INFO] -- [INFO] Total time: 01:42 min [INFO] Finished at: 2017-02-24T08:38:48-08:00 {code} Building C++ code using buck is very fast. We should leverage the agility of C++ in docker environment.
    label: code-design
13. w.r.t. Makefile, on docker the library we link with is under /usr/lib/jvm/java-8-openjdk-amd64/jre/lib/amd64/server where /usr/lib/jvm/java-8-openjdk-amd64/ is JAVA_HOME On Mac, JAVA_HOME is /Library/Java/JavaVirtualMachines/jdk1.8.0_91.jdk/Contents/Home But there is no /Library/Java/JavaVirtualMachines/jdk1.8.0_91.jdk/Contents/Home/jre/lib/amd64/server directory.
14. Patch v14: adds flags to Makefile changes calling convention w.r.t. (*env). replaces char buf allocation with std::string Adding DestroyJavaVM() call resulted in SIGSEGV - commented out for now.
15. Patch v17 is updated with commit 7d284f6b55f5209e86c8a4917215e9218a4c409a (HBASE-17629) Modified client-test.cc so that TestUtil is constructed in SetUpTestCase() The following passed: buck test //core:filter-test buck test //core:client-test
16. Performed "make" under hbase-native-client directory. The build passed.
17. **body:** Thanks Ted for the updated patch. - No commented out code please. - Even if the include directories are different, we cannot hard code the java location like this in the buck and make builds. - We are using c++14 and smart pointers exclusively unless there is a reason not to. So below: {code} + MiniCluster* mini; {code} should be a {{std::unique_ptr<MiniCluster>}} and constructed via {{std::make_unique<MiniCluster>()}}; {{JNIEnv* env}} are exceptions since it is interacting with the C code. But we can also encapsulate those with unique_ptr's after construction. - Indentation is off for some of the methods in mini-cluster.cc. You can run bin/format-code.sh to auto-correct the format. - {{WriteConf}} I think writing conf to a file, and reading it back should not be necessary at all. Unlike the java counterpart, Configuration is much simpler in the cpp side. You can either load it via HBaseConfigurationLoader which reads from the XML files, or you can manually create a Configuration object which does not know anything about XML. You can then just populate relevant properties from the Java configuration. So you should be able to iterate via the Java Conf object, and set values in the cpp Conf object without having to go through the file serialization. - We still need to destroy the JVM after we are done. Were you able to find out the segfault issues? - Rename some of these according to the conventions: {code} + void setup(); -> Setup() + jobject getHTU(); -> htu() // field accessor methods can be named with lower case. + JNIEnv* getEnv(); -> env() + jbyteArray str_to_byte_array(char *str); -> StrToByteChar() + jobject getAdmin(); -> admin {code} All fields of class should have names ending with {{_}}. Like {{putCtor_}}, {{env_}}, etc. - Remove these debug statements? {code} LOG(INFO) {code}
    label: code-design
18. **body:** Patch v18: reference JAVA_HOME in BUCK apply formatter change method names to be Camel case
    label: code-design
19. **body:** I am also getting a lot of these warnings: {code} core/filter-test.cc:65:39: warning: ISO C++11 does not allow conversion from string literal to 'char *' [-Wwritable-strings] FilterTest::test_util_->CreateTable("t", "d"); {code} Also this fails: {code} buck test --all {code}
    label: code-design
20. Patch v20: fixes location-cache-test puts back DestroyJavaVM() call uses $JAVA_HOME in Makefile

21. Patch v22 drops the use of HBASE_CONF environment variable from filter-test.cc and location-cache-test.cc
22. Are you gonna address the remaining review comments? The command {{buck test --all}} still fails, BTW.
23. {code} PASS <100ms 2 Passed 0 Skipped 0 Failed //connection:connection-pool-test PASS <100ms 3 Passed 0 Skipped 0 Failed //serde:client-deserializer-test PASS <100ms 4 Passed 0 Skipped 0 Failed //serde:client-serializer-test PASS <100ms 1 Passed 0 Skipped 0 Failed //serde:region-info-deserializer-test PASS <100ms 4 Passed 0 Skipped 0 Failed //serde:server-name-test PASS <100ms 3 Passed 0 Skipped 0 Failed //serde:table-name-test PASS <100ms 3 Passed 0 Skipped 0 Failed //serde:zk-deserializer-test PASS <100ms 1 Passed 0 Skipped 0 Failed //utils:user-util-test PASS <100ms 7 Passed 0 Skipped 0 Failed //core:cell-test PASS 3.8s 6 Passed 0 Skipped 0 Failed //core:client-test PASS <100ms 5 Passed 0 Skipped 0 Failed //core:configuration-test PASS <100ms 3 Passed 0 Skipped 0 Failed //core:get-test PASS 247ms 17 Passed 0 Skipped 0 Failed //core:hbase_configuration-test PASS <100ms 2 Passed 0 Skipped 0 Failed //core:request_converter-test PASS <100ms 2 Passed 0 Skipped 0 Failed //core:result-test PASS <100ms 5 Passed 0 Skipped 0 Failed //core:scan-test PASS <100ms 2 Passed 0 Skipped 0 Failed //core:time_range-test RESULTS FOR //core:location-cache-test PASS 5.1s 3 Passed 0 Skipped 0 Failed //core:location-cache-test TESTS PASSED PASS 2.7s 2 Passed 0 Skipped 0 Failed //core:filter-test {code} Which test(s) failed for you ?
24. It is failing like this: {code} ====STANDARD ERR==== PASS <100ms 3 Passed 0 Skipped 0 Failed //core:get-test PASS <100ms 17 Passed 0 Skipped 0 Failed //core:hbase_configuration-test FAIL <100ms 0 Passed 0 Skipped 1 Failed //core:location-cache-test FAILURE //core:location-cache-test main: test program aborted before finishing ====STANDARD OUT==== Running main() from gmock_main.cc [==========] Running 3 tests from 1 test case. [----------] Global test environment set-up. [----------] 3 tests from LocationCacheTest WARNING: Logging before InitGoogleLogging() is written to STDERR I0228 20:25:37.822675 30820 mini-cluster.cc:57] set clspath 0 I0228 20:25:37.890406 30820 mini-cluster.cc:77] Launched JVM! -Djava.class.path=/usr/src/buck/build/bootstrapper/bootstrapper.jar:/home/enis/.m2/repository/aopalliance/aopalliance/1.0/aopalliance-1.0.jar:/home/enis/.m2/repository/asm/asm/3.1/asm-3.1.jar:/home/enis/.m2/repository/com/github/stephenc/findbugs/findbugs-annotations/1.3.9-1/findbugs-annotations-1.3.9-1.jar:/home/enis/.m2/repository/com/google/code/findbugs/jsr305/1.3.9/jsr305-1.3.9.jar:/home/enis/.m2/repository/com/google/code/gson/gson/2.2.4/gson-2.2.4.jar:/home/enis/.m2/repository/com/google/guava/guava/12.0.1/guava-12.0.1.jar:/home/enis/.m2/repository/com/google/inject/guice/3.0/guice-3.0.jar:/home/enis/.m2/repository/com/google/inject/extensions/guice-servlet/3.0/guice-servlet-3.0.jar:/home/enis/.m2/repository/com/google/protobuf/protobuf-java/2.5.0/protobuf-java-2.5.0.jar:/home/enis/.m2/repository/com/jamesmurty/utils/java-xmlbuilder/0.4/java-xmlbuilder-0.4.jar:/home/enis/.m2/repository/com/jcraft/jsch/0.1.42/jsch-0.1.42.jar:/home/enis/.m2/repository/com/lmax/disruptor/3.3.0/disruptor-3.3.0.jar:/home/enis/.m2/repository/com/sun/jersey/jersey-client/1.9/jersey-client-1.9.jar:/home/enis/.m2/repository/com/sun/jersey/jersey-core/1.9/jersey-core-1.9.jar:/home/enis/.m2/repository/com/sun/jersey/jersey-json/1.9/jersey-json-1.9.jar:/home/enis/.m2/repository/com/sun/jersey/jersey-server/1.9/jersey-server-1.9.jar:/home/enis/.m2/repository/com/sun/jersey/contribs/jersey-guice/1.9/jersey-guice-1.9.jar:/home/enis/.m2/repository/com/sun/xml/bind/jaxb-impl/2.2.3-1/jaxb-impl-2.2.3-1.jar:/home/enis/.m2/repository/com/thoughtworks/paranamer/paranamer/2.3/paranamer-2.3.jar:/home/enis/.m2/repository/commons-beanutils/commons-beanutils/1.7.0/commons-beanutils-1.7.0.jar:/home/enis/.m2/repository/commons-beanutils/commons-beanutils-core/1.8.3/commons-beanutils-core-1.8.3.jar:/home/enis/.m2/repository/commons-cli/commons-cli/1.2/commons-cli-1.2.jar:/home/enis/.m2/repository/commons-codec/commons-codec/1.9/commons-codec-1.9.jar:/home/enis/.m2/repository/commons-collections/commons-collections/3.2.2/commons-collections-3.2.2.jar:/home/enis/.m2/repository/commons-configuration/commons-configuration/1.6/commons-configuration-1.6.jar:/home/enis/.m2/repository/commons-daemon/commons-daemon/1.0.13/commons-daemon-1.0.13.jar:/home/enis/.m2/repository/commons-digester/commons-digester/1.8/commons-digester-1.8.jar:/home/enis/.m2/repository/commons-el/commons-el/1.0/commons-el-1.0.jar:/home/enis/.m2/repository/commons-fileupload/commons-fileupload/1.3.1/commons-fileupload-1.3.1.jar:/home/enis/.m2/repository/commons-httpclient/commons-httpclient/3.1/commons-httpclient-3.1.jar:/home/enis/.m2/repository/commons-io/commons-io/2.4/commons-io-2.4.jar:/home/enis/.m2/repository/commons-lang/commons-lang/2.6/commons-lang-2.6.jar:/home/enis/.m2/repository/commons-logging/commons-logging/1.2/commons-logging-1.2.jar:/home/enis/.m2/repository/commons-net/commons-net/3.1/commons-net-3.1.jar:/home/enis/.m2/repository/io/dropwizard/metrics/metrics-core/3.1.2/metrics-core-3.1.2.jar:/home/enis/.m2/repository/io/netty/netty-all/4.1.1.Final/netty-all-4.1.1.Final.jar:/home/enis/.m2/repository/javax/activation/activation/1.1/activation-1.1.jar:/home/enis/.m2/repository/javax/inject/javax.inject/1/javax.inject-1.jar:/home/enis/.m2/repository/javax/servlet/servlet-api/2.5/servlet-api-2.5.jar:/home/enis/.m2/repository/javax/xml/bind/jaxb-api/2.2.2/jaxb-api-2.2.2.jar:/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.102-1.b14.el7_2.x86_64/jre/../lib/tools.jar:/home/enis/.m2/repository/junit/junit/4.12/junit-4.12.jar:/home/enis/.m2/repository/log4j/log4j/1.2.17/log4j-1.2.17.jar:/home/enis/.m2/repository/net/java/dev/jets3t/jets3t/0.9.0/jets3t-0.9.0.jar:/home/enis/.m2/repository/net/sourceforge/nekohtml/nekohtml/1.9.16/nekohtml-1.9.16.jar:/home/enis/.m2/repository/org/apache/avro/avro/1.7.6/avro-1.7.6.jar:/home/enis/.m2/repository/org/apache/commons/commons-compress/1.4.1/commons-compress-1.4.1.jar:/home/enis/.m2/repository/org/apache/commons/commons-math/2.2/commons-math-2.2.jar:/home/enis/.m2/repository/org/apache/commons/commons-math3/3.1.1/commons-math3-3.1.1.jar:/home/enis/.m2/repository/org/apache/curator/curator-client/2.7.1/curator-client-2.7.1.jar:/home/enis/.m2/repository/org/apache/curator/curator-framework/2.7.1/curator-framework-2.7.1.jar:/home/enis/.m2/repository/org/apache/curator/curator-recipes/2.7.1/curator-recipes-2.7.1.jar:/home/enis/.m2/repository/org/apache/directory/api/api-asn1-api/1.0.0-M20/api-asn1-api-1.0.0-M20.jar:/home/enis/.m2/repository/org/apache/directory/api/api-util/1.0.0-M20/api-util-1.0.0-M20.jar:/home/enis/.m2/repository/org/apache/directory/server/apacheds-i18n/2.0.0-M15/apacheds-i18n-2.0.0-M15.jar:/home/enis/.m2/repository/org/apache/directory/server/apacheds-kerberos-codec/2.0.0-M15/apacheds-kerberos-codec-2.0.0-M15.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-annotations/2.7.1/hadoop-annotations-2.7.1.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-auth/2.7.1/hadoop-auth-2.7.1.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-client/2.7.1/hadoop-client-2.7.1.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-common/2.7.1/hadoop-common-

```
2.7.1.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-common/2.7.1/hadoop-common-2.7.1-
tests.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-hdfs/2.7.1/hadoop-hdfs-
2.7.1.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-hdfs/2.7.1/hadoop-hdfs-2.7.1-
tests.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-mapreduce-client-app/2.7.1/hadoop-mapreduce-client-app-
2.7.1.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-mapreduce-client-common/2.7.1/hadoop-mapreduce-client-
common-2.7.1.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-mapreduce-client-core/2.7.1/hadoop-mapreduce-client-
core-2.7.1.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-mapreduce-client-hs/2.7.1/hadoop-mapreduce-client-hs-
2.7.1.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-mapreduce-client-jobclient/2.7.1/hadoop-mapreduce-client-
jobclient-2.7.1.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-mapreduce-client-jobclient/2.7.1/hadoop-mapreduce-
client-jobclient-2.7.1-tests.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-mapreduce-client-shuffle/2.7.1/hadoop-
mapreduce-client-shuffle-2.7.1.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-minicluster/2.7.1/hadoop-minicluster-
2.7.1.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-yarn-api/2.7.1/hadoop-yarn-api-
2.7.1.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-yarn-client/2.7.1/hadoop-yarn-client-
2.7.1.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-yarn-common/2.7.1/hadoop-yarn-common-
2.7.1.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-yarn-server-applicationhistoryservice/2.7.1/hadoop-yarn-server-
applicationhistoryservice-2.7.1.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-yarn-server-common/2.7.1/hadoop-yarn-
server-common-2.7.1.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-yarn-server-nodemanager/2.7.1/hadoop-yarn-
server-nodemanager-2.7.1.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-yarn-server-resourcemanager/2.7.1/hadoop-
yarn-server-resourcemanager-2.7.1.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-yarn-server-tests/2.7.1/hadoop-yarn-
server-tests-2.7.1-tests.jar:/home/enis/.m2/repository/org/apache/hadoop/hadoop-yarn-server-web-proxy/2.7.1/hadoop-yarn-server-
web-proxy-2.7.1.jar:/home/enis/projects/hbase/hbase-annotations/target/hbase-annotations-2.0.0-
SNAPSHOT.jar:/home/enis/projects/hbase/hbase-annotations/target/hbase-annotations-2.0.0-SNAPSHOT-
tests.jar:/home/enis/projects/hbase/hbase-client/target/hbase-client-2.0.0-SNAPSHOT.jar:/home/enis/projects/hbase/hbase-
common/target/hbase-common-2.0.0-SNAPSHOT.jar:/home/enis/projects/hbase/hbase-common/target/hbase-common-2.0.0-
SNAPSHOT-tests.jar:/home/enis/projects/hbase/hbase-external-blockcache/target/hbase-external-blockcache-2.0.0-
SNAPSHOT.jar:/home/enis/projects/hbase/hbase-hadoop-compat/target/hbase-hadoop-compat-2.0.0-
SNAPSHOT.jar:/home/enis/projects/hbase/hbase-hadoop-compat/target/hbase-hadoop-compat-2.0.0-SNAPSHOT-
tests.jar:/home/enis/projects/hbase/hbase-hadoop2-compat/target/hbase-hadoop2-compat-2.0.0-
SNAPSHOT.jar:/home/enis/projects/hbase/hbase-hadoop2-compat/target/hbase-hadoop2-compat-2.0.0-SNAPSHOT-
tests.jar:/home/enis/projects/hbase/hbase-it/target/hbase-it-2.0.0-SNAPSHOT-tests.jar:/home/enis/projects/hbase/hbase-prefix-
tree/target/hbase-prefix-tree-2.0.0-SNAPSHOT.jar:/home/enis/projects/hbase/hbase-procedure/target/hbase-procedure-2.0.0-
SNAPSHOT.jar:/home/enis/projects/hbase/hbase-protocol/target/hbase-protocol-2.0.0-
SNAPSHOT.jar:/home/enis/projects/hbase/hbase-resource-bundle/target/hbase-resource-bundle-2.0.0-
SNAPSHOT.jar:/home/enis/projects/hbase/hbase-rest/target/hbase-rest-2.0.0-SNAPSHOT.jar:/home/enis/projects/hbase/hbase-
rsgroup/target/hbase-rsgroup-2.0.0-SNAPSHOT.jar:/home/enis/projects/hbase/hbase-server/target/hbase-server-2.0.0-
SNAPSHOT.jar:/home/enis/projects/hbase/hbase-server/target/hbase-server-2.0.0-SNAPSHOT-
tests.jar:/home/enis/projects/hbase/hbase-shell/target/hbase-shell-2.0.0-SNAPSHOT.jar:/home/enis/projects/hbase/hbase-
spark/target/hbase-spark-2.0.0-SNAPSHOT.jar:/home/enis/projects/hbase/hbase-testing-util/target/hbase-testing-util-2.0.0-
SNAPSHOT.jar:/home/enis/projects/hbase/hbase-thrift/target/hbase-thrift-2.0.0-
SNAPSHOT.jar:/home/enis/.m2/repository/org/apache/htrace/htrace-core/3.1.0-incubating/htrace-core-3.1.0-
incubating.jar:/home/enis/.m2/repository/org/apache/httpcomponents/httpclient/4.3.6/httpclient-
4.3.6.jar:/home/enis/.m2/repository/org/apache/httpcomponents/httpcore/4.4.4/httpcore-
4.4.4.jar:/home/enis/.m2/repository/org/apache/thrift/libthrift/0.9.3/libthrift-
0.9.3.jar:/home/enis/.m2/repository/org/apache/xmlgraphics/batik-css/1.8/batik-css-
1.8.jar:/home/enis/.m2/repository/org/apache/xmlgraphics/batik-ext/1.8/batik-ext-
1.8.jar:/home/enis/.m2/repository/org/apache/xmlgraphics/batik-util/1.8/batik-util-
1.8.jar:/home/enis/.m2/repository/org/apache/zookeeper/zookeeper/3.4.8/zookeeper-
3.4.8.jar:/home/enis/.m2/repository/org/apache/zookeeper/zookeeper/3.4.6/zookeeper-3.4.6-
tests.jar:/home/enis/.m2/repository/org/beanshell/bsh-core/2.0b4/bsh-core-
2.0b4.jar:/home/enis/.m2/repository/org/codehaus/jackson/jackson-core-asl/1.9.13/jackson-core-asl-
1.9.13.jar:/home/enis/.m2/repository/org/codehaus/jackson/jackson-jaxrs/1.9.13/jackson-jaxrs-
1.9.13.jar:/home/enis/.m2/repository/org/codehaus/jackson/jackson-mapper-asl/1.9.13/jackson-mapper-asl-
1.9.13.jar:/home/enis/.m2/repository/org/codehaus/jackson/jackson-xc/1.9.13/jackson-xc-
1.9.13.jar:/home/enis/.m2/repository/org/codehaus/jettison/jettison/1.3.3/jettison-
1.3.3.jar:/home/enis/.m2/repository/org/fusesource/leveldbjni/leveldbjni-all/1.8/leveldbjni-all-
1.8.jar:/home/enis/.m2/repository/org/hamcrest/hamcrest-core/1.3/hamcrest-core-
1.3.jar:/home/enis/.m2/repository/org/jamon/jamon-runtime/2.4.1/jamon-runtime-
2.4.1.jar:/home/enis/.m2/repository/org/jruby/jruby-complete/1.6.8/jruby-complete-
1.6.8.jar:/home/enis/.m2/repository/org/jruby/jcodings/jcodings/1.0.8/jcodings-
1.0.8.jar:/home/enis/.m2/repository/org/jruby/joni/joni/2.1.2/joni-2.1.2.jar:/home/enis/.m2/repository/org/mockito/mockito-
all/1.10.8/mockito-all-1.10.8.jar:/home/enis/.m2/repository/org/mortbay/jetty/jetty/6.1.26/jetty-
6.1.26.jar:/home/enis/.m2/repository/org/mortbay/jetty/jetty-sslengine/6.1.26/jetty-sslengine-
6.1.26.jar:/home/enis/.m2/repository/org/mortbay/jetty/jetty-util/6.1.26/jetty-util-
6.1.26.jar:/home/enis/.m2/repository/org/mortbay/jetty/jsp-2.1/6.1.14/jsp-2.1-
6.1.14.jar:/home/enis/.m2/repository/org/mortbay/jetty/jsp-api-2.1/6.1.14/jsp-api-2.1-
6.1.14.jar:/home/enis/.m2/repository/org/mortbay/jetty/servlet-api-2.5/6.1.14/servlet-api-2.5-
6.1.14.jar:/home/enis/.m2/repository/org/owasp/antisamy/antisamy/1.5.3/antisamy-
1.5.3.jar:/home/enis/.m2/repository/org/owasp/esapi/esapi/2.1.0.1/esapi-2.1.0.1.jar:/home/enis/.m2/repository/org/slf4j/slf4j-
api/1.7.7/slf4j-api-1.7.7.jar:/home/enis/.m2/repository/org/slf4j/slf4j-log4j12/1.7.7/slf4j-log4j12-
1.7.7.jar:/home/enis/.m2/repository/org/tukaani/xz/1.0/xz-1.0.jar:/home/enis/.m2/repository/org/xerial/snappy/snappy-
java/1.0.5/snappy-java-1.0.5.jar:/home/enis/.m2/repository/tomcat/jasper-compiler/5.5.23/jasper-compiler-
5.5.23.jar:/home/enis/.m2/repository/tomcat/jasper-runtime/5.5.23/jasper-runtime-
```

5.5.23.jar:/home/enis/.m2/repository/xalan/xalan/2.7.0/xalan-2.7.0.jar:/home/enis/.m2/repository/xerces/xercesImpl/2.9.1/xercesImpl-2.9.1.jar:/home/enis/.m2/repository/xml-apis/xml-apis/1.3.03/xml-apis-1.3.03.jar:/home/enis/.m2/repository/xml-apis/xml-apis-ext/1.3.04/xml-apis-ext-1.3.04.jar:/home/enis/.m2/repository/xmlenc/xmlenc/0.52/xmlenc-0.52.jar:/home/enis/.m2/repository/xom/xom/1.2.5/xom-1.2.5.jar I0228 20:25:37.915246 30820 mini-cluster.cc:123] Couldn't find class HBaseTestingUtility ====STANDARD ERR==== PASS <100ms 2 Passed 0 Skipped 0 Failed //core:request_converter-test PASS <100ms 2 Passed 0 Skipped 0 Failed //core:result-test PASS <100ms 5 Passed 0 Skipped 0 Failed //core:scan-test PASS <100ms 2 Passed 0 Skipped 0 Failed //core:time_range-test TESTS FAILED: 3 FAILURES Failed target: //core:filter-test FAIL //core:filter-test Failed target: //core:location-cache-test FAIL //core:location-cache-test Failed target: //core:client-test FAIL //core:client-test {code} Somehow the HBaseTestingUtility cannot be loaded from the classpath (see the above error message).

25. Nevermind, I think it is due to the {{mvn install}} command being run from outside the docker (because .m2 paths are not docker paths). Let me build inside the docker.
26. I was able to run the tests successfully. I checked the valgrind output with client-test like this: {code} valgrind buck-out/gen/core/client-test 2>&1 | tee val.out {code} It is showing thousands of illegal memory accesses for the JVM now. I am not sure whether these are related to how we pass the arguments.
27. After installing valgrind, I got (using above valgrind command): {code} valgrind: the 'impossible' happened: LibVEX called failure_exit(). host stacktrace: ==65== at 0x38083F48: ??? (in /usr/lib/valgrind/memcheck-amd64-linux) ==65== by 0x38084064: ??? (in /usr/lib/valgrind/memcheck-amd64-linux) {code} Looks like I hit: https://bugs.launchpad.net/ubuntu/+source/valgrind/+bug/1574437
28. w.r.t. WriteConf(), looking at configuration.h, there is no equivalent method in Configuration class. WriteConf() would be useful for HBaseConfigurationLoader class (reading persisted hbase-site.xml).
29. **body:** Noticed the following from valgrind output: {code} ==21686== Invalid write of size 4 ==21686== at 0x91BBBCC: ??? ==21686== by 0x91AFE53: ??? ==21686== by 0x91AFFFC: ??? ==21686== by 0x91AFFFC: ??? ==21686== by 0x91A84E6: ??? ==21686== by 0x5497D7A: ??? (in /usr/lib/jvm/java-8-openjdk-amd64/jre/lib/amd64/server/libjvm.so) ==21686== by 0x5496643: ??? (in /usr/lib/jvm/java-8-openjdk-amd64/jre/lib/amd64/server/libjvm.so) ==21686== by 0x5496C07: ??? (in /usr/lib/jvm/java-8-openjdk-amd64/jre/lib/amd64/server/libjvm.so) ==21686== by 0x584007D: ??? (in /usr/lib/jvm/java-8-openjdk-amd64/jre/lib/amd64/server/libjvm.so) ==21686== by 0x54BEE41: JNI_CreateJavaVM (in /usr/lib/jvm/java-8-openjdk-amd64/jre/lib/amd64/server/libjvm.so) ==21686== by 0x426260: hbase::MiniCluster::CreateVM(JavaVM_**) (mini-cluster.cc:73) ==21686== by 0x426B5A: hbase::MiniCluster::Setup() (mini-cluster.cc:115) ==21686== Address 0xffefeb710 is on thread 1's stack ==21686== 81920 bytes below stack pointer {code} Line 73 is this line: {code} rv = JNI_CreateJavaVM(jvm, (void **)&env_, &args); {code} Where jvm is JavaVM **jvm Without symbols from libjvm.so, it is not easy to know what the actual error is.
    label: code-design
30. Patch v23 moves the creation of conf dir to MiniCluster
31. **body:** I've checked the v23 patch. You did not address these previous comments: - There is still a very large number of these kind of warnings: {code} core/filter-test.cc:65:39: warning: ISO C++11 does not allow conversion from string literal to 'char *' [-Wwritable-strings] FilterTest::test_util_->CreateTable("t", "d"); {code} You should use std::string everywhere, except when you are sending it to the JVM object. So, the method signatures like: {code} jobject MiniCluster::CreateTable(char *tblNam, char *familyName) { {code} should be always taking std::string, or better {{const std::string&}}. We want to minimize the {{char *}} usage to be only happening in the JVM layer. You can use string::c_str(). - Also this warning: {code} test-util/mini-cluster.cc:260:1: warning: control reaches end of non-void function [-Wreturn-type] {code} - Some of the fields in MiniCluster does not end with {{_}}. Examples are {{cluster}}, {{jvm}}, etc. - As per above, you should not need WriteConf() method at all. Please construct a Configuration object and populate it via calling {{Configuration::Set()}} from the java-level Configuration object. You seem to be doing that in TestUtil, so please remove the WriteConf and setting the path, etc. Also return std::string in GetConf(), etc. - We use {{make lint}} to run the cpplint script. Can you please address these new warnings as well: {code} test-util/test-util.cc:65: Add #include <memory> for make_unique<> [build/include_what_you_use] [4] test-util/test-util.cc:77: Could not find a newline character at the end of the file. [whitespace/ending_newline] [5] Done processing test-util/test-util.cc test-util/test-util.h:84: Add #include <memory> for shared_ptr<> [build/include_what_you_use] [4] Done processing test-util/test-util.h test-util/mini-cluster.cc:22: "glog/logging.h" already included at test-util/mini-cluster.cc:21 [build/include] [4] test-util/mini-cluster.cc:25: Found C system header after C++ system header. Should be: mini-cluster.h, c system, c++ system, other. [build/include_order] [4] test-util/mini-cluster.cc:26: Found C system header after C++ system header. Should be: mini-cluster.h, c system, c++ system, other. [build/include_order] [4] test-util/mini-cluster.cc:70: Missing space before { [whitespace/braces] [5] test-util/mini-cluster.cc:75: Using C-style cast. Use reinterpret_cast<void **>(...) instead [readability/casting] [4] test-util/mini-cluster.cc:227: Using C-style cast. Use reinterpret_cast<jbyte *>(...) instead [readability/casting] [4] test-util/mini-cluster.cc:313: Add #include <string> for string [build/include_what_you_use] [4] Done processing test-util/mini-cluster.cc test-util/mini-cluster.h:28: Include the directory when naming .h files [build/include] [4] test-util/mini-cluster.h:35: Add #include <string> for string [build/include_what_you_use] [4] {code} - GetConf() is overloaded. For the one returning a configuration value, you should name it GetConfValue(). - Should the jobject cluster be a field in MiniCluster. No need to pass it to methods like GetConf() I think. In StartCluster(), you can save it in the field. - This behavior coming from the existing code: {code} Creating a TestUtil will spin up a cluster with numRegionServers region servers. {code} But I think it is wrong. Can we change it so that TestUtil ctor does not start the cluster, but you have to manually call StartCluster. I think it is better this way, because in theory you should be able use the test util without the cluster. Other then these, the patch looks good.
    label: code-design
32. w.r.t. lint warnings, there are a lot of existing ones. e.g. {code} serde/region-info.h:24: Found C system header after other header. Should be: region-info.h, c system, c++ system, other. [build/include_order] [4] {code} I am trying to get rid of ones in new files.
33. bq. This behavior coming from the existing code How about addressing the last comment in another issue (since it is pre-existing behavior) ?
34. Patch v26 addresses Enis' latest comments. WriteConf() is kept for core:client-test
35. **body:** Patch v27 drops unneeded jobject return type. RunShellCmd() is also removed since it is no longer used. All tests pass.
    label: code-design
36. Ran valgrind again. libjvm.so still showed up but there was no occurrence of MiniCluster. This was from last run: {code} ==21686== by 0x426260: hbase::MiniCluster::CreateVM(JavaVM_**) (mini-cluster.cc:73) ==21686== by 0x426B5A:

hbase::MiniCluster::Setup() (mini-cluster.cc:115) {code}

37. **body:** Patch v28 removes lint warnings introduced by the previous patch.
    label: code-design
38. **body:** Can you please remove all commented out code. There are still references. Also, in multiple places, you should remove these comments as well: {code} // The connection stays open and we don't want that. // So we are stopping the connection. // We can remove this once we have fixed the folly part - delete test_util; {code}
    label: documentation
39. **body:** Patch v29 removes commented out code.
    label: documentation
40. Rebased patch. The following passed: buck test //core:retry-test
41. Patch v31 replaces the camel case variable names in MiniCluster.
42. "buck test --all" passed for patch v31.
43. **body:** Thanks Ted for the changes. - Please remove ALL commented out code. This is the third time for the same comment. {code} + //hbase::Configuration conf; + // hbase::TestUtil *test_util = new hbase::TestUtil(2, ClientTest::kDefHBaseConfPath.c_str()); {code} - Refactor {{addColMid_}} -> {{add_col_mid_}}. and {{setConfMid_}}. - These methods should take {{const std::string&}} instead of {{char *}}. {code} {{void WriteConf(jobject conf, const char *filepath);}} +jbyteArray MiniCluster::StrToByteChar(const char *str) { +TestUtil::TestUtil(int servers, const char *confPath) {code} - Remove: {code} + LOG(INFO) << "retrieving " << key; + LOG(INFO) << "Got string " << val; {code} and maybe: {code} + LOG(INFO) << "retrieved port " << port; {code} - Change these to check for NULL instead: {code} if (put == 0) { + if (mid == 0) { + if (n == 0) return NULL; {code} - I'll do the changes for client-test for not depending on the WriteConf() as a follow up.
    label: code-design
44. Patch v33 addresses review comments above. All tests pass.
45. Remove this: {code} - hbase::Configuration conf; + //hbase::Configuration conf; {code} Please commit to branch afterwards. Thanks.
46. Thanks for the review, Enis.