ACCUMULO-3643 Catch findbugs up through rank 16  
\* fixes incorrect clone implementations  
\* fix missing equals/hashCode for Comparable  
\* public non-final static constants in ClassSize  
\* unclosed streams  
\* inefficient boxing/unboxing  
\* incorrectly using read instead of readFully  
\* check return on file operations (delete, create, mkdirs, etc.)  
\* add comments to indicate intentional ignoring exceptions  
\* renamed an inner class which looked like an exception, but wasn't  
\* make servlets conform to serializable interface from parent class  
\* triage and add exclusions for System.exit and forced gc  
\* triage and add exclusions for unclosed stream false positives  
  
Project: accumulo.txt  
Commit: 8158d39c9dc9efb980db66051c7a01b45a880a9a  
ClassName: org.apache.accumulo.harness.MiniClusterHarness.java  
  
@@ -177,3 +179,3 @@ protected void configureForSsl(MiniAccumuloConfigImpl cfg, File folder) {  
 File sslDir = new File(folder, "ssl");  
- sslDir.mkdirs();  
**+ assertTrue(sslDir.mkdirs() || sslDir.isDirectory());** File rootKeystoreFile = new File(sslDir, "root-" + cfg.getInstanceName() + ".jks");  
========================================

ACCUMULO-1965 Fix exception handling for namespaces  
Make the exception handling propagate correctly to the client.  
Vastly expanded on existing namespace tests to include checks for  
throwing the correct exceptions. Consolidated a lot of the fate  
operations code and refactored the master (slightly; moved inner  
classes to separate files) to be easier to modify the relevant RPC  
handling code.  
Fixed many API bugs related to throwing correct exceptions, found from  
the new tests added to NamespacesIT.  
  
Project: accumulo.txt  
Commit: f35e3f47265868c560b3d49edebf1a8c24196512  
ClassName: org.apache.accumulo.core.client.admin.TableOperationsImpl.java  
  
@@ -1342,6 +1327,6 @@ public void offline(String tableName, boolean wait) throws AccumuloSecurityExcep  
 try {  
- doTableOperation(TableOperation.OFFLINE, args, opts);  
+ doTableFateOperation(tableName, TableNotFoundException.class, FateOperation.TABLE\_OFFLINE, args, opts);  
 } catch (TableExistsException e) {  
- throw new RuntimeException(e);  
**+ throw new AssertionError(e);** }  
========================================

ACCUMULO-3529 ACCUMULO-3493 Test numerous tprocolfactory for proxy  
Finds and fixes a regression about the protocol factory used in the proxy.  
Should verify that the protocol factory configured for use is actually used.  
  
Project: accumulo.txt  
Commit: 8aada370d316c56313251a178af99e51c95db610  
ClassName: org.apache.accumulo.proxy.SimpleProxyBase.java  
  
+ public void testInstanceOperations() throws Exception {  
+ int tservers = 0;  
+ for (String tserver : client.getTabletServers(creds)) {  
+ client.pingTabletServer(creds, tserver);  
+ tservers++;  
+ }  
**+ assertTrue(tservers > 0);**+ Map<String,String> cfg = client.getSiteConfiguration(creds);  
**+ assertTrue(cfg.get("instance.dfs.dir").startsWith(macTestFolder.getPath()));**+ client.setProperty(creds, "table.split.threshold", "500M");  
+ for (int i = 0; i < 5; i++) {  
+ cfg = client.getSystemConfiguration(creds);  
+ if ("500M".equals(cfg.get("table.split.threshold")))  
+ break;  
+ UtilWaitThread.sleep(200);  
+ }  
**+ assertEquals("500M", cfg.get("table.split.threshold"));**+ client.removeProperty(creds, "table.split.threshold");  
+ for (int i = 0; i < 5; i++) {  
+ cfg = client.getSystemConfiguration(creds);  
+ if (!"500M".equals(cfg.get("table.split.threshold")))  
+ break;  
+ UtilWaitThread.sleep(200);  
+ }  
**+ assertNotEquals("500M", cfg.get("table.split.threshold"));  
+ assertTrue(client.testClassLoad(creds, DevNull.class.getName(), SortedKeyValueIterator.class.getName()));  
+ assertFalse(client.testClassLoad(creds, "foo.bar", SortedKeyValueIterator.class.getName()));**+ client.createTable(creds, "slow", true, TimeType.MILLIS);  
+ IteratorSetting setting = new IteratorSetting(100, "slow", SlowIterator.class.getName(), Collections.singletonMap("sleepTime", "250"));  
+ client.attachIterator(creds, "slow", setting, EnumSet.allOf(IteratorScope.class));  
+ for (int i = 0; i < 40; i++) {  
+ client.updateAndFlush(creds, "slow", mutation("row" + i, "cf", "cq", "value"));  
+ }  
+ Thread t = new Thread() {  
+ @Override  
+ public void run() {  
+ String scanner;  
+ try {  
+ Client client2 = new TestProxyClient("localhost", proxyPort, getProtocol()).proxy();  
+ scanner = client2.createScanner(creds, "slow", null);  
+ client2.nextK(scanner, 10);  
+ client2.closeScanner(scanner);  
+ } catch (Exception e) {  
+ throw new RuntimeException(e);  
+ }  
+ }  
+ };  
+ t.start();  
+ List<ActiveScan> scans = new ArrayList<ActiveScan>();  
+ for (int i = 0; i < 100 && scans.isEmpty(); i++) {  
+ for (String tserver : client.getTabletServers(creds)) {  
+ List<ActiveScan> scansForServer = client.getActiveScans(creds, tserver);  
+ for (ActiveScan scan : scansForServer) {  
+ if ("root".equals(scan.getUser())) {  
+ scans.add(scan);  
+ }  
+ }  
+ if (!scans.isEmpty())  
+ break;  
+ UtilWaitThread.sleep(100);  
+ }  
+ }  
+ t.join();  
**+ assertFalse(scans.isEmpty());**+ boolean found = false;  
+ Map<String,String> map = null;  
+ for (int i = 0; i < scans.size() && !found; i++) {  
+ ActiveScan scan = scans.get(i);  
+ if ("root".equals(scan.getUser())) {  
**+ assertTrue(ScanState.RUNNING.equals(scan.getState()) || ScanState.QUEUED.equals(scan.getState()));  
+ assertEquals(ScanType.SINGLE, scan.getType());  
+ assertEquals("slow", scan.getTable());**+ map = client.tableIdMap(creds);  
**+ assertEquals(map.get("slow"), scan.getExtent().tableId);  
+ assertTrue(scan.getExtent().endRow == null);  
+ assertTrue(scan.getExtent().prevEndRow == null);**+ found = true;  
+ }  
+ }  
**+ assertTrue("Could not find a scan against the 'slow' table", found);**+ t = new Thread() {  
+ @Override  
+ public void run() {  
+ try {  
+ Client client2 = new TestProxyClient("localhost", proxyPort, getProtocol()).proxy();  
+ client2.compactTable(creds, "slow", null, null, null, true, true, null);  
+ } catch (Exception e) {  
+ throw new RuntimeException(e);  
+ }  
+ }  
+ };  
+ t.start();  
+ final String desiredTableId = map.get("slow");  
+ List<ActiveCompaction> compactions = new ArrayList<ActiveCompaction>();  
+ for (int i = 0; i < 100 && compactions.isEmpty(); i++) {  
+ for (String tserver : client.getTabletServers(creds)) {  
+ List<ActiveCompaction> compactionsOnServer = client.getActiveCompactions(creds, tserver);  
+ for (ActiveCompaction compact : compactionsOnServer) {  
+ if (desiredTableId.equals(compact.getExtent().tableId)) {  
+ compactions.add(compact);  
+ }  
+ }  
+ if (!compactions.isEmpty())  
+ break;  
+ }  
+ UtilWaitThread.sleep(10);  
+ }  
+ t.join();  
**+ assertFalse(compactions.isEmpty());**+ for (ActiveCompaction c : compactions) {  
+ if (desiredTableId.equals(c.getExtent().tableId)) {  
**+ assertTrue(c.inputFiles.isEmpty());  
+ assertEquals(CompactionType.MINOR, c.getType());  
+ assertEquals(CompactionReason.USER, c.getReason());  
+ assertEquals("", c.localityGroup);  
+ assertTrue(c.outputFile.contains("default\_tablet"));**+ return;  
+ }  
+ }  
+ fail("Expection to find running compaction for table 'slow' but did not find one");  
+ }  
========================================

ACCUMULO-1965 Fix exception handling for namespaces  
Make the exception handling propagate correctly to the client.  
Vastly expanded on existing namespace tests to include checks for  
throwing the correct exceptions. Consolidated a lot of the fate  
operations code and refactored the master (slightly; moved inner  
classes to separate files) to be easier to modify the relevant RPC  
handling code.  
Fixed many API bugs related to throwing correct exceptions, found from  
the new tests added to NamespacesIT.  
  
Project: accumulo.txt  
Commit: f35e3f47265868c560b3d49edebf1a8c24196512  
ClassName: org.apache.accumulo.core.client.admin.TableOperationsImpl.java  
  
@@ -668,6 +674,6 @@ public void delete(String tableName) throws AccumuloException, AccumuloSecurityE  
 try {  
- doTableOperation(TableOperation.DELETE, args, opts);  
+ doTableFateOperation(tableName, TableNotFoundException.class, FateOperation.TABLE\_DELETE, args, opts);  
 } catch (TableExistsException e) {  
- throw new RuntimeException(e);  
**+ throw new AssertionError(e);** }  
========================================

ACCUMULO-3643 Catch findbugs up through rank 16  
\* fixes incorrect clone implementations  
\* fix missing equals/hashCode for Comparable  
\* public non-final static constants in ClassSize  
\* unclosed streams  
\* inefficient boxing/unboxing  
\* incorrectly using read instead of readFully  
\* check return on file operations (delete, create, mkdirs, etc.)  
\* add comments to indicate intentional ignoring exceptions  
\* renamed an inner class which looked like an exception, but wasn't  
\* make servlets conform to serializable interface from parent class  
\* triage and add exclusions for System.exit and forced gc  
\* triage and add exclusions for unclosed stream false positives  
  
Project: accumulo.txt  
Commit: 8158d39c9dc9efb980db66051c7a01b45a880a9a  
ClassName: org.apache.accumulo.core.client.security.tokens.NullToken.java  
  
 public NullToken clone() {  
- return new NullToken();  
+ try {  
+ return (NullToken) super.clone();  
+ } catch (CloneNotSupportedException e) {  
**+ throw new AssertionError("Inconceivable", e);**+ }  
========================================

ACCUMULO-3147 Create replication table at initialize, upgrade  
Migrate to all replication code to a built-in system-managed replication  
table instead of the user-space one. Ensure all replication-related tests  
pass. Also fix a small bug in Mock which threw an exception when using  
deleteRows with null start and end rows.  
  
Project: accumulo.txt  
Commit: e840549a9c6376c8c5dbe9663e14b4987c10dc5e  
ClassName: org.apache.accumulo.core.replication.ReplicationTable.java  
  
- public static BatchWriter getBatchWriter(Connector conn) throws TableNotFoundException {  
+ public static BatchWriter getBatchWriter(Connector conn) throws ReplicationTableOfflineException {  
+ try {  
 return conn.createBatchWriter(NAME, new BatchWriterConfig());  
+ } catch (TableNotFoundException e) {  
**+ throw new AssertionError(NAME + " should exist, but doesn't.");**+ } catch (TableOfflineException e) {  
+ throw new ReplicationTableOfflineException(e);  
+ }  
 }  
========================================

ACCUMULO-1965 Fix exception handling for namespaces  
Make the exception handling propagate correctly to the client.  
Vastly expanded on existing namespace tests to include checks for  
throwing the correct exceptions. Consolidated a lot of the fate  
operations code and refactored the master (slightly; moved inner  
classes to separate files) to be easier to modify the relevant RPC  
handling code.  
Fixed many API bugs related to throwing correct exceptions, found from  
the new tests added to NamespacesIT.  
  
Project: accumulo.txt  
Commit: f35e3f47265868c560b3d49edebf1a8c24196512  
ClassName: org.apache.accumulo.core.client.admin.TableOperationsImpl.java  
  
@@ -543,6 +549,6 @@ public void deleteRows(String tableName, Text start, Text end) throws AccumuloEx  
 try {  
- doTableOperation(TableOperation.DELETE\_RANGE, args, opts);  
+ doTableFateOperation(tableName, TableNotFoundException.class, FateOperation.TABLE\_DELETE\_RANGE, args, opts);  
 } catch (TableExistsException e) {  
- throw new RuntimeException(e);  
**+ throw new AssertionError(e);** }  
========================================

ACCUMULO-1965 Fix exception handling for namespaces  
Make the exception handling propagate correctly to the client.  
Vastly expanded on existing namespace tests to include checks for  
throwing the correct exceptions. Consolidated a lot of the fate  
operations code and refactored the master (slightly; moved inner  
classes to separate files) to be easier to modify the relevant RPC  
handling code.  
Fixed many API bugs related to throwing correct exceptions, found from  
the new tests added to NamespacesIT.  
  
Project: accumulo.txt  
Commit: f35e3f47265868c560b3d49edebf1a8c24196512  
ClassName: org.apache.accumulo.core.client.admin.NamespaceOperationsImpl.java  
  
+ private String doNamespaceFateOperation(FateOperation op, List<ByteBuffer> args, Map<String,String> opts) throws AccumuloSecurityException,  
+ AccumuloException, NamespaceExistsException, NamespaceNotFoundException {  
+ try {  
+ return tableOps.doFateOperation(op, args, opts);  
+ } catch (TableExistsException e) {  
**+ throw new AssertionError(e);**+ } catch (TableNotFoundException e) {  
**+ throw new AssertionError(e);**+ }  
+ }  
========================================

ACCUMULO-3147 Create replication table at initialize, upgrade  
Migrate to all replication code to a built-in system-managed replication  
table instead of the user-space one. Ensure all replication-related tests  
pass. Also fix a small bug in Mock which threw an exception when using  
deleteRows with null start and end rows.  
  
Project: accumulo.txt  
Commit: e840549a9c6376c8c5dbe9663e14b4987c10dc5e  
ClassName: org.apache.accumulo.core.replication.ReplicationTable.java  
  
- public static Scanner getScanner(Connector conn) throws TableNotFoundException {  
+ public static Scanner getScanner(Connector conn) throws ReplicationTableOfflineException {  
+ try {  
 return conn.createScanner(NAME, Authorizations.EMPTY);  
+ } catch (TableNotFoundException e) {  
**+ throw new AssertionError(NAME + " should exist, but doesn't.");**+ } catch (TableOfflineException e) {  
+ throw new ReplicationTableOfflineException(e);  
+ }  
 }  
========================================

ACCUMULO-1579 Fix classpath issues and add plugin test with iterator  
  
Project: accumulo.txt  
Commit: 3d7a6e71abea53485ef37b9e6e27134a722c6088  
ClassName: org.apache.accumulo.plugin.PluginIT.java  
  
- public static void tearDown() throws Exception {}  
+ @Test  
+ public void checkIterator() throws IOException, AccumuloException, AccumuloSecurityException, TableExistsException, TableNotFoundException {  
+ String tableName = "checkIterator";  
+ connector.tableOperations().create(tableName);  
+ BatchWriter bw = connector.createBatchWriter(tableName, new BatchWriterConfig());  
+ Mutation m = new Mutation("ROW1");  
+ m.put("allowed", "CQ1", "V1");  
+ m.put("denied", "CQ2", "V2");  
+ m.put("allowed", "CQ3", "V3");  
+ bw.addMutation(m);  
+ m = new Mutation("ROW2");  
+ m.put("allowed", "CQ1", "V1");  
+ m.put("denied", "CQ2", "V2");  
+ m.put("allowed", "CQ3", "V3");  
+ bw.addMutation(m);  
+ bw.close();  
+ Scanner scanner = connector.createScanner(tableName, Authorizations.EMPTY);  
+ IteratorSetting is = new IteratorSetting(5, CustomFilter.class);  
+ scanner.addScanIterator(is);  
+ int count = 0;  
+ for (Entry<Key,Value> entry : scanner) {  
+ count++;  
**+ assertEquals("allowed", entry.getKey().getColumnFamily().toString());**+ }  
**+ assertEquals(4, count);**+ scanner.clearScanIterators();  
+ CustomFilter.setNegate(is, true);  
+ scanner.addScanIterator(is);  
+ count = 0;  
+ for (Entry<Key,Value> entry : scanner) {  
+ count++;  
**+ assertEquals("denied", entry.getKey().getColumnFamily().toString());**+ }  
**+ assertEquals(2, count);  
+ assertTrue(new File("target/accumulo-maven-plugin/" + instance.getInstanceName() + "/testCheckIteratorPassed").createNewFile());**+ }  
========================================

https://issues.apache.org/jira/browse/AMQ-4757 activemq-jms-pool a generic jms xa pool derived from activemq-pool which activemq-pool now extends with amq specifics  
  
Project: activemq.txt  
Commit: b66559ee079bd55bfc49daae6b95b79706147fbe  
ClassName: org.apache.activemq.jms.pool.PooledConnection.java  
  
**+ protected void assertNotClosed() throws javax.jms.IllegalStateException {**+ if (stopped || pool == null) {  
+ throw new IllegalStateException("Connection closed");  
+ }  
+ }  
========================================

https://issues.apache.org/jira/browse/AMQ-5294  
This closes #36  
  
Project: activemq.txt  
Commit: e47e0e046350a8325ef5948c2c12b0dc4f053916  
ClassName: org.apache.activemq.tool.JmsProducerSystem.java  
  
+ protected ClientRunBasis getClientRunBasis() {  
**+ assert (producer != null);**+ return ClientRunBasis.valueOf(producer.getSendType().toLowerCase());  
+ }  
+ @Override  
+ protected long getClientRunDuration() {  
+ return producer.getSendDuration();  
========================================

Applied patch for https://issues.apache.org/activemq/browse/AMQ-1925  
  
Project: activemq.txt  
Commit: 9ce91fa62503f45af4828e528a3c799853cff3bc  
ClassName: org.apache.activemq.broker.region.PrefetchSubscription.java  
  
@@ -412,2 +412,41 @@ public void afterRollback()  
**+ protected void assertAckMatchesDispatched(MessageAck ack)**+ throws JMSException {  
+ MessageId firstAckedMsg = ack.getFirstMessageId();  
+ MessageId lastAckedMsg = ack.getLastMessageId();  
+ int checkCount = 0;  
+ boolean checkFoundStart = false;  
+ boolean checkFoundEnd = false;  
+ for (MessageReference node : dispatched) {  
+ if (!checkFoundStart && firstAckedMsg != null && firstAckedMsg.equals(node.getMessageId())) {  
+ checkFoundStart = true;  
+ }  
+ if (checkFoundStart || firstAckedMsg == null)  
+ checkCount++;  
+ if (lastAckedMsg != null && lastAckedMsg.equals(node.getMessageId())) {  
+ checkFoundEnd = true;  
+ break;  
+ }  
+ }  
+ if (!checkFoundStart && firstAckedMsg != null)  
+ throw new JMSException("Unmatched acknowledege: Could not find Message-ID "+firstAckedMsg+" in dispatched-list (start of ack)");  
+ if (!checkFoundEnd && lastAckedMsg != null)  
+ throw new JMSException("Unmatched acknowledege: Could not find Message-ID "+firstAckedMsg+" in dispatched-list (end of ack)");  
+ if (ack.getMessageCount() != checkCount) {  
+ throw new JMSException("Unmatched acknowledege: Expected message count ("+ack.getMessageCount()+  
+ ") differs from count in dispatched-list ("+checkCount+")");  
+ }  
+ }  
========================================

fix https://issues.apache.org/jira/browse/AMQ-4280  
activemq-core module renamed to activemq-unit-tests  
  
Project: activemq.txt  
Commit: ef24cc9a045da960161fd0ee9f6839b2164e9333  
ClassName: org.apache.activemq.broker.region.cursors.CursorSupport.java  
  
+ public void testSendWhilstConsume() throws Exception {  
+ ConnectionFactory factory = createConnectionFactory();  
+ Connection consumerConnection = getConsumerConnection(factory);  
+ MessageConsumer consumer = getConsumer(consumerConnection);  
+ consumerConnection.close();  
+ Connection producerConnection = factory.createConnection();  
+ producerConnection.start();  
+ Session session = producerConnection.createSession(false, Session.AUTO\_ACKNOWLEDGE);  
+ MessageProducer producer = session.createProducer(getDestination(session));  
+ List<TextMessage> senderList = new ArrayList<TextMessage>();  
+ for (int i = 0; i < MESSAGE\_COUNT / 10; i++) {  
+ TextMessage msg = session.createTextMessage("test" + i);  
+ senderList.add(msg);  
+ producer.send(msg);  
+ }  
+ consumerConnection = getConsumerConnection(factory);  
+ consumer = getConsumer(consumerConnection);  
+ final List<Message> consumerList = new ArrayList<Message>();  
+ final CountDownLatch latch = new CountDownLatch(1);  
+ consumer.setMessageListener(new MessageListener() {  
+ public void onMessage(Message msg) {  
+ try {  
+ Thread.sleep(50);  
+ } catch (Exception e) {  
+ e.printStackTrace();  
+ }  
+ consumerList.add(msg);  
+ if (consumerList.size() == MESSAGE\_COUNT) {  
+ latch.countDown();  
+ }  
+ }  
+ });  
+ for (int i = MESSAGE\_COUNT / 10; i < MESSAGE\_COUNT; i++) {  
+ TextMessage msg = session.createTextMessage("test" + i);  
+ senderList.add(msg);  
+ producer.send(msg);  
+ }  
+ latch.await(300000, TimeUnit.MILLISECONDS);  
+ producerConnection.close();  
+ consumerConnection.close();  
**+ assertEquals("Still dipatching - count down latch not sprung", latch.getCount(), 0);**+ for (int i = 0; i < senderList.size(); i++) {  
+ Message sent = senderList.get(i);  
+ Message consumed = consumerList.get(i);  
+ if (!sent.equals(consumed)) {  
+ LOG.error("BAD MATCH AT POS " + i);  
+ LOG.error(sent.toString());  
+ LOG.error(consumed.toString());  
+ }  
**+ assertEquals("This should be the same at pos " + i + " in the list", sent.getJMSMessageID(), consumed.getJMSMessageID());**+ }  
+ }  
========================================

http://issues.apache.org/activemq/browse/AMQ-907  
  
Project: activemq.txt  
Commit: e2bd38c7a2e3b78b2ff0c770864ce04e2a6d5ff1  
ClassName: org.apache.activemq.transport.activeio.ActiveIOTransport.java  
  
+ public Response request(Command command,int timeout) throws IOException {  
**+ throw new AssertionError("Unsupported Method");**+ }  
========================================

Fix for https://issues.apache.org/jira/browse/AMQ-3621 - Integrate Apache Shiro with ActiveMQ as "security solution" - many thanks to Les Hazlewood (lhazlewood) for the patch!  
Commit dev changes to branch.  
Signed-off-by: Christian Posta <christian.posta@gmail.com>  
Added shiro support to 5.9.x  
Signed-off-by: Christian Posta <christian.posta@gmail.com>  
Added Shiro deps to dist zip/tarball and 'all' .jar  
Signed-off-by: Christian Posta <christian.posta@gmail.com>  
updated version number to reflect correctly packaging Shiro features  
Signed-off-by: Christian Posta <christian.posta@gmail.com>  
Handled case when adding a producer might not have a specified destination.  
Signed-off-by: Christian Posta <christian.posta@gmail.com>  
Merged latest Connection API changes from trunk.  
Signed-off-by: Christian Posta <christian.posta@gmail.com>  
removed accidental addition of Stormpath's m2 repo  
Signed-off-by: Christian Posta <christian.posta@gmail.com>  
reverted accidental deletion of <distributionManagement>  
Signed-off-by: Christian Posta <christian.posta@gmail.com>  
  
Project: activemq.txt  
Commit: f9451e56e20f8d1712ed09c579d493feee514898  
ClassName: org.apache.activemq.shiro.authz.AuthorizationFilter.java  
  
**+ protected void assertAuthorized(DestinationAction action) {  
+ assertAuthorized(action, action.getVerb());**+ }  
========================================

Depending on the test configuration parameters, it was possible to get an OutOfMemory error. The causes were:  
- The MessageList was holding on to all the messages being consumed, changed this so that it only holds on to the messageIds  
- Was using a non persistent broker, but was sending it persistent messages, in the topic case, he holds on to the messages in a memory based message store. By default we now send non persistent messages.  
  
Project: activemq.txt  
Commit: 3d95025a477b1d2e27e29880883f13a1cd37d047  
ClassName: org.apache.activemq.util.MessageIdList.java  
  
**+ public void assertMessagesReceivedNoWait(int messageCount) {  
+ assertEquals("expected number of messages when received", messageCount, getMessageCount());**+ }  
========================================

fix https://issues.apache.org/jira/browse/AMQ-4280  
activemq-core module renamed to activemq-unit-tests  
  
Project: activemq.txt  
Commit: ef24cc9a045da960161fd0ee9f6839b2164e9333  
ClassName: org.apache.activemq.util.MessageIdList.java  
  
**+ public void assertMessagesReceived(int messageCount) {**+ waitForMessagesToArrive(messageCount);  
**+ assertMessagesReceivedNoWait(messageCount);**+ }  
========================================

fix https://issues.apache.org/jira/browse/AMQ-4280  
activemq-core module renamed to activemq-unit-tests  
  
Project: activemq.txt  
Commit: ef24cc9a045da960161fd0ee9f6839b2164e9333  
ClassName: org.apache.activemq.util.MessageIdList.java  
  
**+ public void assertAtMostMessagesReceived(int messageCount) {**+ int actual = getMessageCount();  
**+ assertTrue("at most: " + messageCount + " messages received. Actual: " + actual, actual <= messageCount);**+ }  
========================================

https://issues.apache.org/jira/browse/AMQ-5306  
This closes #39  
  
Project: activemq.txt  
Commit: 6dd47bb63fb37a4d1ff491188e75f613b6b81935  
ClassName: org.apache.activemq.tool.AbstractJmsClient.java  
  
+ private String[] mapToSimpleNames(String[] destNames) {  
**+ assert (destNames != null);**+ String[] simpleNames = new String[destNames.length];  
+ for (int i = 0; i < destNames.length; i++) {  
+ simpleNames[i] = getSimpleName(destNames[i]);  
+ }  
+ return simpleNames;  
+ }  
========================================

Applying colins patch https://issues.apache.org/activemq/browse/AMQ-2244  
Thanks!  
  
Project: activemq-apollo.txt  
Commit: 8241083d86bc5579eca80f78ff18ed23b8026991  
ClassName: org.apache.activemq.queue.SharedQueue.java  
  
+ public void elementRemoved(QueueElement qe) {  
+ if (!loaded && qe.sequence < recoverySequence && qe.sequence > lastRecoverdSequence) {  
+ return;  
+ }  
+ if (qe.isExpirable()) {  
+ count--;  
+ removeFromCache(qe);  
**+ assert count > 0;**+ }  
+ }  
========================================

Applying patch at https://issues.apache.org/activemq/browse/AMQ-2255  
  
Project: activemq-apollo.txt  
Commit: 3131481436673062ee49c69a4ab120f373af671e  
ClassName: org.apache.activemq.queue.ExclusivePersistentQueue.java  
  
+ public void add(E elem, ISourceController<?> source) {  
+ synchronized (this) {  
**+ assert initialized;**+ controller.add(elem, source);  
+ accepted(source, elem);  
+ }  
+ }  
========================================

Applying patch at https://issues.apache.org/activemq/browse/AMQ-2255  
  
Project: activemq-apollo.txt  
Commit: 3131481436673062ee49c69a4ab120f373af671e  
ClassName: org.apache.activemq.queue.CursoredQueue.java  
  
+ public final void releaseHardRef(IFlowController<QueueElement<V>> controller) {  
+ hardRefs--;  
+ if (hardRefs == 0) {  
+ unload(controller);  
+ }  
+ if (controller != null) {  
+ controller.elementDispatched(this);  
+ }  
**+ assert hardRefs >= 0;**+ }  
========================================

Applying Colins patch at: https://issues.apache.org/activemq/browse/AMQ-2115 basically:  
Along with some other refactoring, these changes are an incremental step towards support a non blocking i/o environment conducive to single/low thread count broker.  
Listing of changes:  
1. Introduced org.apache.activemq.transport.DispatchableTransport and DispatchableTransportServer. These can be provided a dispatcher and operate in a non blocking fashion. They need to be backed by a FlowController to avoid overflow.  
2. Changed PipeTransportFactory to operate in non blocking fashion by implementing above interfaces.  
3. Added copy of org.apache.activemq.transport.nio package which are being refactored so they can be used with above interfaces.  
4. Modified RemoteProducer, RemoteConsumer and BrokerConnection to extend RemoteConnection to take advantage of common network level changes described below.  
**5. Changed RemoteConnection to incorporate network WindowLimiter which assert flow control via protocol instead of relying on transport level flow control. Also updated RemoteConnection to take a Dispatcher. Deleted AbstractTestConnection.**6. Changed FlowController and PriorityFlowController not to implement IFlowSink. Instead introduced org.apache.activemq.SingleFlowRelay which wraps the FlowController and acts as a Sink/Source that does not queue messages that it is passing through, this now acts as the output sink for RemoteConnection.  
7. Modified test.proto ProtocolBuffers definition to include new FlowControl message used to communicate space released by the flow controller in support of using protocol based flowcontrol instead of transport based flow control.  
8. Introduced org.apache.activemq.wireformat.StatefulWireformat which allows stateful non blocking marshalling/unmarshalling. Changed ProtoWireFormatFactory.TestWireformat to implement this so that non blocking i/o can be achieved.  
  
Project: activemq-apollo.txt  
Commit: ef430cdcee7f047bc6c3821c4af86c0ce96c5e51  
ClassName: org.apache.activemq.transport.nio.NIOTransport.java  
  
+ private void serviceRead() {  
+ try {  
+ while (true) {  
+ int readSize = channel.read(currentBuffer);  
+ if (readSize == -1) {  
+ onException(new EOFException());  
+ selection.close();  
+ break;  
+ }  
+ if (readSize == 0) {  
+ break;  
+ }  
+ if (currentBuffer.hasRemaining()) {  
+ continue;  
+ }  
+ if (nextFrameSize == -1) {  
**+ assert inputBuffer == currentBuffer;**+ inputBuffer.flip();  
+ nextFrameSize = inputBuffer.getInt() + 4;  
+ if (nextFrameSize > inputBuffer.capacity()) {  
+ currentBuffer = ByteBuffer.allocate(nextFrameSize);  
+ currentBuffer.putInt(nextFrameSize);  
+ } else {  
+ inputBuffer.limit(nextFrameSize);  
+ }  
+ } else {  
+ currentBuffer.flip();  
+ Object command = wireFormat.unmarshal(new DataInputStream(new NIOInputStream(currentBuffer)));  
+ doConsume((Command)command);  
+ nextFrameSize = -1;  
+ inputBuffer.clear();  
+ inputBuffer.limit(4);  
+ currentBuffer = inputBuffer;  
+ }  
+ }  
+ } catch (IOException e) {  
+ onException(e);  
+ } catch (Throwable e) {  
+ onException(IOExceptionSupport.create(e));  
+ }  
+ }  
========================================

Applying colins patch https://issues.apache.org/activemq/browse/AMQ-2244  
Thanks!  
  
Project: activemq-apollo.txt  
Commit: 8241083d86bc5579eca80f78ff18ed23b8026991  
ClassName: org.apache.activemq.queue.SharedQueue.java  
  
+ public final void releaseSoftRef() {  
+ softRefs--;  
+ if (softRefs == 0) {  
+ unload(null);  
+ }  
**+ assert softRefs >= 0;** }  
========================================

Applying colins patch https://issues.apache.org/activemq/browse/AMQ-2244  
Thanks!  
  
Project: activemq-apollo.txt  
Commit: 8241083d86bc5579eca80f78ff18ed23b8026991  
ClassName: org.apache.activemq.queue.SharedQueue.java  
  
- public final void releaseCursorRef(Cursor cursor) {  
- cursorRefs--;  
- if (cursorRefs == 0) {  
- unload(cursor.memoryController);  
+ public final void releaseHardRef(IFlowController<QueueElement> controller) {  
+ hardRefs--;  
+ if (hardRefs == 0) {  
+ unload(controller);  
 }  
+ if (controller != null) {  
+ controller.elementDispatched(this);  
+ }  
**+ assert hardRefs >= 0;**+ }  
========================================

Applying patch at https://issues.apache.org/activemq/browse/AMQ-2255  
  
Project: activemq-apollo.txt  
Commit: 3131481436673062ee49c69a4ab120f373af671e  
ClassName: org.apache.activemq.queue.CursoredQueue.java  
  
+ public final void releaseSoftRef() {  
+ softRefs--;  
+ if (softRefs == 0) {  
+ unload(null);  
+ }  
**+ assert softRefs >= 0;**+ }  
========================================

- Better perf report  
- fixed tcp receive buffer array out of bounds exception.  
  
Project: activemq-apollo.txt  
Commit: f7fb587316a52f60d9852cd78a574e988b03fedd  
ClassName: org.apache.activemq.transport.tcp.TcpTransport.java  
  
@@ -487,2 +487,7 @@ private void drainInbound() throws IOException {  
 Object command = this.wireformat.unmarshalNB(readBuffer);  
**+ assert wireformat.unmarshalStartPos() <= wireformat.unmarshalEndPos();  
+ assert wireformat.unmarshalEndPos() <= readBuffer.position();** if (command != null) {  
========================================

Applying patch at https://issues.apache.org/activemq/browse/AMQ-2255  
  
Project: activemq-apollo.txt  
Commit: 3131481436673062ee49c69a4ab120f373af671e  
ClassName: org.apache.activemq.queue.ExclusivePersistentQueue.java  
  
+ public boolean offer(E elem, ISourceController<?> source) {  
+ synchronized (this) {  
**+ assert initialized;**+ if (controller.offer(elem, source)) {  
+ accepted(source, elem);  
+ return true;  
+ }  
+ return false;  
+ }  
+ }  
========================================

**Add a linked assertion check to provide a friendlier usage error message.**  
Project: activemq-apollo.txt  
Commit: 4e164765ff7189cf345c2053f7de0a45df77c935  
ClassName: main.scala.org.apache.activemq.apollo.util.list.LinkedNode.java  
  
**+ final private void assertLinked() {  
+ assert list!=null && !list.isEmpty() : "Node is not linked";**+ }  
 final public boolean isHeadNode() {  
**+ assertLinked();**========================================

ACTIVEMQ6-1 - Initial HornetQ Donation Commit  
https://issues.apache.org/jira/browse/ACTIVEMQ6-1  
This is the initial donation of the HornetQ codebase as per document http://incubator.apache.org/ip-clearance/hornetq.html  
  
Project: activemq-artemis.txt  
Commit: 8ecd255f981e5d806173cf86f6fac9c4c7c25167  
ClassName: org.hornetq.core.server.impl.PostOfficeJournalLoader.java  
  
+ public void recoverPendingPageCounters(List<PageCountPending> pendingNonTXPageCounter) throws Exception  
+ {  
+ Transaction txRecoverCounter = new TransactionImpl(storageManager);  
+ Map<SimpleString, Map<Long, Map<Long, List<PageCountPending>>>> perAddressMap = generateMapsOnPendingCount(queues, pendingNonTXPageCounter, txRecoverCounter);  
+ for (SimpleString address : perAddressMap.keySet())  
+ {  
+ PagingStore store = pagingManager.getPageStore(address);  
+ Map<Long, Map<Long, List<PageCountPending>>> perPageMap = perAddressMap.get(address);  
**+ assert (perPageMap != null);**+ for (Long pageId : perPageMap.keySet())  
+ {  
+ Map<Long, List<PageCountPending>> perQueue = perPageMap.get(pageId);  
**+ assert (perQueue != null);**+ if (store.checkPageFileExists(pageId.intValue()))  
+ {  
+ Page pg = store.createPage(pageId.intValue());  
+ pg.open();  
+ List<PagedMessage> pgMessages = pg.read(storageManager);  
+ Map<Long, AtomicInteger> countsPerQueueOnPage = new HashMap<Long, AtomicInteger>();  
+ for (PagedMessage pgd : pgMessages)  
+ {  
+ if (pgd.getTransactionID() <= 0)  
+ {  
+ for (long q : pgd.getQueueIDs())  
+ {  
+ AtomicInteger countQ = countsPerQueueOnPage.get(q);  
+ if (countQ == null)  
+ {  
+ countQ = new AtomicInteger(0);  
+ countsPerQueueOnPage.put(q, countQ);  
+ }  
+ countQ.incrementAndGet();  
+ }  
+ }  
+ }  
+ for (Map.Entry<Long, List<PageCountPending>> entry : perQueue.entrySet())  
+ {  
+ for (PageCountPending record : entry.getValue())  
+ {  
+ HornetQServerLogger.LOGGER.debug("Deleting pg tempCount " + record.getID());  
+ storageManager.deletePendingPageCounter(txRecoverCounter.getID(), record.getID());  
+ }  
+ PageSubscriptionCounter counter = store.getCursorProvider().getSubscription(entry.getKey()).getCounter();  
+ AtomicInteger value = countsPerQueueOnPage.get(entry.getKey());  
+ if (value == null)  
+ {  
+ HornetQServerLogger.LOGGER.debug("Page " + entry.getKey() + " wasn't open, so we will just ignore");  
+ }  
+ else  
+ {  
+ HornetQServerLogger.LOGGER.debug("Replacing counter " + value.get());  
+ counter.increment(txRecoverCounter, value.get());  
+ }  
+ }  
+ }  
+ else  
+ {  
+ HornetQServerLogger.LOGGER.debug("Page " + pageId + " didn't exist on address " + address + ", so we are just removing records");  
+ for (List<PageCountPending> records : perQueue.values())  
+ {  
+ for (PageCountPending record : records)  
+ {  
+ HornetQServerLogger.LOGGER.debug("Removing pending page counter " + record.getID());  
+ storageManager.deletePendingPageCounter(txRecoverCounter.getID(), record.getID());  
+ txRecoverCounter.setContainsPersistent();  
+ }  
+ }  
+ }  
+ }  
+ }  
+ txRecoverCounter.commit();  
+ }  
========================================

ACTIVEMQ6-1 - Initial HornetQ Donation Commit  
https://issues.apache.org/jira/browse/ACTIVEMQ6-1  
This is the initial donation of the HornetQ codebase as per document http://incubator.apache.org/ip-clearance/hornetq.html  
  
Project: activemq-artemis.txt  
Commit: 8ecd255f981e5d806173cf86f6fac9c4c7c25167  
ClassName: org.hornetq.jms.client.ThreadAwareContext.java  
  
**+ public void assertNotMessageListenerThreadRuntime()**+ {  
+ if (messageListenerThreads.contains(Thread.currentThread().getId()))  
+ {  
+ throw HornetQJMSClientBundle.BUNDLE.callingMethodFromListenerRuntime();  
+ }  
+ }  
========================================

ACTIVEMQ6-3 renaming directories from activemq6 to activemq  
https://issues.apache.org/jira/browse/ACTIVEMQ6-3  
We are renaming packages from activemq6 to activemq as that's more generic and version independent  
On this first commit I'm just renaming the directories otherwise the history would be lost. The next commit will rename the text on the directories.  
If I squash these two commits git will make us delete / add again.  
  
Project: activemq-artemis.txt  
Commit: 3c44e268548d91b703abc96b0002aed8abcc9668  
ClassName: org.apache.activemq.core.paging.cursor.impl.PageCursorProviderImpl.java  
  
+ public void cleanup()  
+ {  
+ ArrayList<Page> depagedPages = new ArrayList<Page>();  
+ while (true)  
+ {  
+ if (pagingStore.lock(100))  
+ {  
+ break;  
+ }  
+ if (!pagingStore.isStarted())  
+ return;  
+ }  
+ synchronized (this)  
+ {  
+ try  
+ {  
+ if (!pagingStore.isStarted())  
+ {  
+ return;  
+ }  
+ if (pagingStore.getNumberOfPages() == 0)  
+ {  
+ return;  
+ }  
+ if (HornetQServerLogger.LOGGER.isDebugEnabled())  
+ {  
**+ HornetQServerLogger.LOGGER.debug("Asserting cleanup for address " + this.pagingStore.getAddress());**+ }  
+ ArrayList<PageSubscription> cursorList = cloneSubscriptions();  
+ long minPage = checkMinPage(cursorList);  
+ if (minPage == pagingStore.getCurrentWritingPage() && pagingStore.getCurrentPage().getNumberOfMessages() > 0)  
+ {  
+ boolean complete = true;  
+ for (PageSubscription cursor : cursorList)  
+ {  
+ if (!cursor.isComplete(minPage))  
+ {  
+ if (HornetQServerLogger.LOGGER.isDebugEnabled())  
+ {  
+ HornetQServerLogger.LOGGER.debug("Cursor " + cursor + " was considered incomplete at page " + minPage);  
+ }  
+ complete = false;  
+ break;  
+ }  
+ else  
+ {  
+ if (HornetQServerLogger.LOGGER.isDebugEnabled())  
+ {  
+ HornetQServerLogger.LOGGER.debug("Cursor " + cursor + "was considered \*\*complete\*\* at page " + minPage);  
+ }  
+ }  
+ }  
+ if (!pagingStore.isStarted())  
+ {  
+ return;  
+ }  
+ if (complete)  
+ {  
+ if (HornetQServerLogger.LOGGER.isDebugEnabled())  
+ {  
+ HornetQServerLogger.LOGGER.debug("Address " + pagingStore.getAddress() +  
+ " is leaving page mode as all messages are consumed and acknowledged from the page store");  
+ }  
+ pagingStore.forceAnotherPage();  
+ Page currentPage = pagingStore.getCurrentPage();  
+ storeBookmark(cursorList, currentPage);  
+ pagingStore.stopPaging();  
+ }  
+ }  
+ for (long i = pagingStore.getFirstPage(); i < minPage; i++)  
+ {  
+ Page page = pagingStore.depage();  
+ if (page == null)  
+ {  
+ break;  
+ }  
+ depagedPages.add(page);  
+ }  
+ if (pagingStore.getNumberOfPages() == 0 || pagingStore.getNumberOfPages() == 1 &&  
+ pagingStore.getCurrentPage().getNumberOfMessages() == 0)  
+ {  
+ pagingStore.stopPaging();  
+ }  
+ else  
+ {  
+ if (HornetQServerLogger.LOGGER.isTraceEnabled())  
+ {  
+ HornetQServerLogger.LOGGER.trace("Couldn't cleanup page on address " + this.pagingStore.getAddress() +  
+ " as numberOfPages == " +  
+ pagingStore.getNumberOfPages() +  
+ " and currentPage.numberOfMessages = " +  
+ pagingStore.getCurrentPage().getNumberOfMessages());  
+ }  
+ }  
+ }  
+ catch (Exception ex)  
+ {  
+ HornetQServerLogger.LOGGER.problemCleaningPageAddress(ex, pagingStore.getAddress());  
+ return;  
+ }  
+ finally  
+ {  
+ pagingStore.unlock();  
+ }  
+ }  
+ try  
+ {  
+ for (Page depagedPage : depagedPages)  
+ {  
+ PageCache cache;  
+ PagedMessage[] pgdMessages;  
+ synchronized (softCache)  
+ {  
+ cache = softCache.get((long)depagedPage.getPageId());  
+ }  
+ if (isTrace)  
+ {  
+ HornetQServerLogger.LOGGER.trace("Removing page " + depagedPage.getPageId() + " from page-cache");  
+ }  
+ if (cache == null)  
+ {  
+ storageManager.beforePageRead();  
+ List<PagedMessage> pgdMessagesList = null;  
+ try  
+ {  
+ depagedPage.open();  
+ pgdMessagesList = depagedPage.read(storageManager);  
+ }  
+ finally  
+ {  
+ try  
+ {  
+ depagedPage.close();  
+ }  
+ catch (Exception e)  
+ {  
+ }  
+ storageManager.afterPageRead();  
+ }  
+ depagedPage.close();  
+ pgdMessages = pgdMessagesList.toArray(new PagedMessage[pgdMessagesList.size()]);  
+ }  
+ else  
+ {  
+ pgdMessages = cache.getMessages();  
+ }  
+ depagedPage.delete(pgdMessages);  
+ onDeletePage(depagedPage);  
+ synchronized (softCache)  
+ {  
+ softCache.remove((long)depagedPage.getPageId());  
+ }  
+ }  
+ }  
+ catch (Exception ex)  
+ {  
+ HornetQServerLogger.LOGGER.problemCleaningPageAddress(ex, pagingStore.getAddress());  
+ return;  
+ }  
+ }  
========================================

ACTIVEMQ6-4 - Rename packages to ActiveMQ  
https://issues.apache.org/jira/browse/ACTIVEMQ6-4  
Repackage the modules, java source and maven poms to apache.activemq6  
  
Project: activemq-artemis.txt  
Commit: 23e8edd9791b5a2ac025c321f97a9dd2329bbeaa  
ClassName: org.apache.activemq6.jms.client.HornetQMessageProducer.java  
  
+ public void close() throws JMSException  
+ {  
**+ connection.getThreadAwareContext().assertNotCompletionListenerThread();**+ try  
+ {  
+ clientProducer.close();  
+ }  
+ catch (HornetQException e)  
+ {  
+ throw JMSExceptionHelper.convertFromHornetQException(e);  
+ }  
+ }  
========================================

ACTIVEMQ6-4 - Rename packages to ActiveMQ  
https://issues.apache.org/jira/browse/ACTIVEMQ6-4  
Repackage the modules, java source and maven poms to apache.activemq6  
  
Project: activemq-artemis.txt  
Commit: 23e8edd9791b5a2ac025c321f97a9dd2329bbeaa  
ClassName: org.apache.activemq6.jms.client.ThreadAwareContext.java  
  
**+ public void assertNotMessageListenerThread() throws IllegalStateException**+ {  
+ if (messageListenerThreads.contains(Thread.currentThread().getId()))  
+ {  
+ throw HornetQJMSClientBundle.BUNDLE.callingMethodFromListener();  
+ }  
+ }  
========================================

ACTIVEMQ6-3 renaming directories from activemq6 to activemq  
https://issues.apache.org/jira/browse/ACTIVEMQ6-3  
We are renaming packages from activemq6 to activemq as that's more generic and version independent  
On this first commit I'm just renaming the directories otherwise the history would be lost. The next commit will rename the text on the directories.  
If I squash these two commits git will make us delete / add again.  
  
Project: activemq-artemis.txt  
Commit: 3c44e268548d91b703abc96b0002aed8abcc9668  
ClassName: org.apache.activemq.jms.client.HornetQSession.java  
  
+ public void close() throws JMSException  
+ {  
**+ connection.getThreadAwareContext().assertNotCompletionListenerThread();  
+ connection.getThreadAwareContext().assertNotMessageListenerThread();**+ synchronized (connection)  
+ {  
+ try  
+ {  
+ for (HornetQMessageConsumer cons : new HashSet<HornetQMessageConsumer>(consumers))  
+ {  
+ cons.close();  
+ }  
+ session.close();  
+ connection.removeSession(this);  
+ }  
+ catch (HornetQException e)  
+ {  
+ throw JMSExceptionHelper.convertFromHornetQException(e);  
+ }  
+ }  
+ }  
========================================

ACTIVEMQ6-1 - Initial HornetQ Donation Commit  
https://issues.apache.org/jira/browse/ACTIVEMQ6-1  
This is the initial donation of the HornetQ codebase as per document http://incubator.apache.org/ip-clearance/hornetq.html  
  
Project: activemq-artemis.txt  
Commit: 8ecd255f981e5d806173cf86f6fac9c4c7c25167  
ClassName: org.hornetq.core.server.impl.HornetQServerImpl.java  
  
+ public final synchronized void start() throws Exception  
+ {  
+ if (state != SERVER\_STATE.STOPPED)  
+ {  
+ HornetQServerLogger.LOGGER.debug("Server already started!");  
+ return;  
+ }  
+ synchronized (failbackCheckerGuard)  
+ {  
+ cancelFailBackChecker = false;  
+ }  
+ state = SERVER\_STATE.STARTING;  
+ activationLatch.setCount(1);  
+ HornetQServerLogger.LOGGER.debug("Starting server " + this);  
+ OperationContextImpl.clearContext();  
+ try  
+ {  
+ checkJournalDirectory();  
+ nodeManager =  
+ createNodeManager(configuration.getJournalDirectory(), configuration.getHAPolicy().getBackupGroupName(), false);  
+ nodeManager.start();  
+ HornetQServerLogger.LOGGER.serverStarting((configuration.getHAPolicy().isBackup() ? "backup" : "live"), configuration);  
+ if (configuration.isRunSyncSpeedTest())  
+ {  
+ SyncSpeedTest test = new SyncSpeedTest();  
+ test.run();  
+ }  
+ final boolean wasLive = !configuration.getHAPolicy().isBackup();  
+ if (!configuration.getHAPolicy().isBackup())  
+ {  
+ if (configuration.getHAPolicy().isSharedStore() && configuration.isPersistenceEnabled())  
+ {  
+ activation = new SharedStoreLiveActivation();  
+ }  
+ else  
+ {  
+ activation = new SharedNothingLiveActivation();  
+ }  
+ activation.run();  
+ }  
+ if (configuration.getHAPolicy().isBackup())  
+ {  
+ if (configuration.getHAPolicy().isSharedStore())  
+ {  
+ activation = new SharedStoreBackupActivation();  
+ }  
+ else  
+ {  
**+ assert replicationEndpoint == null;**+ nodeManager.stop();  
+ nodeManager =  
+ createNodeManager(configuration.getJournalDirectory(), configuration.getHAPolicy().getBackupGroupName(), true);  
+ backupUpToDate = false;  
+ backupSyncLatch.setCount(1);  
+ replicationEndpoint = new ReplicationEndpoint(this, shutdownOnCriticalIO, wasLive);  
+ activation = new SharedNothingBackupActivation(wasLive);  
+ }  
+ backupActivationThread = new Thread(activation, HornetQMessageBundle.BUNDLE.activationForServer(this));  
+ backupActivationThread.start();  
+ }  
+ else  
+ {  
+ state = SERVER\_STATE.STARTED;  
+ HornetQServerLogger.LOGGER.serverStarted(getVersion().getFullVersion(), nodeManager.getNodeId(),  
+ identity != null ? identity : "");  
+ }  
+ connectorsService = new ConnectorsService(configuration, storageManager, scheduledPool, postOffice);  
+ connectorsService.start();  
+ }  
+ finally  
+ {  
+ OperationContextImpl.clearContext();  
+ }  
+ }  
========================================

ACTIVEMQ6-3 renaming directories from activemq6 to activemq  
https://issues.apache.org/jira/browse/ACTIVEMQ6-3  
We are renaming packages from activemq6 to activemq as that's more generic and version independent  
On this first commit I'm just renaming the directories otherwise the history would be lost. The next commit will rename the text on the directories.  
If I squash these two commits git will make us delete / add again.  
  
Project: activemq-artemis.txt  
Commit: 3c44e268548d91b703abc96b0002aed8abcc9668  
ClassName: org.apache.activemq.jms.client.HornetQJMSContext.java  
  
+ public void rollback()  
+ {  
**+ threadAwareContext.assertNotCompletionListenerThreadRuntime();**+ checkSession();  
+ try  
+ {  
+ session.rollback();  
+ }  
+ catch (JMSException e)  
+ {  
+ throw JmsExceptionUtils.convertToRuntimeException(e);  
+ }  
+ }  
========================================

ACTIVEMQ6-4 - Rename packages to ActiveMQ  
https://issues.apache.org/jira/browse/ACTIVEMQ6-4  
Repackage the modules, java source and maven poms to apache.activemq6  
  
Project: activemq-artemis.txt  
Commit: 23e8edd9791b5a2ac025c321f97a9dd2329bbeaa  
ClassName: org.apache.activemq6.core.journal.impl.JournalImpl.java  
  
+ private synchronized JournalLoadInformation load(final LoaderCallback loadManager, final boolean changeData,  
+ final JournalState replicationSync) throws Exception  
+ {  
+ if (state == JournalState.STOPPED || state == JournalState.LOADED)  
+ {  
+ throw new IllegalStateException("Journal " + this + " must be in " + JournalState.STARTED + " state, was " +  
+ state);  
+ }  
+ if (state == replicationSync)  
+ {  
+ throw new IllegalStateException("Journal cannot be in state " + JournalState.STARTED);  
+ }  
+ checkControlFile();  
+ records.clear();  
+ filesRepository.clear();  
+ transactions.clear();  
+ currentFile = null;  
+ final Map<Long, TransactionHolder> loadTransactions = new LinkedHashMap<Long, TransactionHolder>();  
+ final List<JournalFile> orderedFiles = orderFiles();  
+ filesRepository.calculateNextfileID(orderedFiles);  
+ int lastDataPos = JournalImpl.SIZE\_HEADER;  
+ final AtomicLong maxID = new AtomicLong(-1);  
+ for (final JournalFile file : orderedFiles)  
+ {  
+ JournalImpl.trace("Loading file " + file.getFile().getFileName());  
+ final AtomicBoolean hasData = new AtomicBoolean(false);  
+ int resultLastPost = JournalImpl.readJournalFile(fileFactory, file, new JournalReaderCallback()  
+ {  
+ private void checkID(final long id)  
+ {  
+ if (id > maxID.longValue())  
+ {  
+ maxID.set(id);  
+ }  
+ }  
+ public void onReadAddRecord(final RecordInfo info) throws Exception  
+ {  
+ checkID(info.id);  
+ hasData.set(true);  
+ loadManager.addRecord(info);  
+ records.put(info.id, new JournalRecord(file, info.data.length + JournalImpl.SIZE\_ADD\_RECORD + 1));  
+ }  
+ public void onReadUpdateRecord(final RecordInfo info) throws Exception  
+ {  
+ checkID(info.id);  
+ hasData.set(true);  
+ loadManager.updateRecord(info);  
+ JournalRecord posFiles = records.get(info.id);  
+ if (posFiles != null)  
+ {  
+ posFiles.addUpdateFile(file, info.data.length + JournalImpl.SIZE\_ADD\_RECORD + 1); // +1 = compact  
+ }  
+ }  
+ public void onReadDeleteRecord(final long recordID) throws Exception  
+ {  
+ hasData.set(true);  
+ loadManager.deleteRecord(recordID);  
+ JournalRecord posFiles = records.remove(recordID);  
+ if (posFiles != null)  
+ {  
+ posFiles.delete(file);  
+ }  
+ }  
+ public void onReadUpdateRecordTX(final long transactionID, final RecordInfo info) throws Exception  
+ {  
+ onReadAddRecordTX(transactionID, info);  
+ }  
+ public void onReadAddRecordTX(final long transactionID, final RecordInfo info) throws Exception  
+ {  
+ checkID(info.id);  
+ hasData.set(true);  
+ TransactionHolder tx = loadTransactions.get(transactionID);  
+ if (tx == null)  
+ {  
+ tx = new TransactionHolder(transactionID);  
+ loadTransactions.put(transactionID, tx);  
+ }  
+ tx.recordInfos.add(info);  
+ JournalTransaction tnp = transactions.get(transactionID);  
+ if (tnp == null)  
+ {  
+ tnp = new JournalTransaction(transactionID, JournalImpl.this);  
+ transactions.put(transactionID, tnp);  
+ }  
+ tnp.addPositive(file, info.id, info.data.length + JournalImpl.SIZE\_ADD\_RECORD\_TX + 1); // +1 = compact  
+ }  
+ public void onReadDeleteRecordTX(final long transactionID, final RecordInfo info) throws Exception  
+ {  
+ hasData.set(true);  
+ TransactionHolder tx = loadTransactions.get(transactionID);  
+ if (tx == null)  
+ {  
+ tx = new TransactionHolder(transactionID);  
+ loadTransactions.put(transactionID, tx);  
+ }  
+ tx.recordsToDelete.add(info);  
+ JournalTransaction tnp = transactions.get(transactionID);  
+ if (tnp == null)  
+ {  
+ tnp = new JournalTransaction(transactionID, JournalImpl.this);  
+ transactions.put(transactionID, tnp);  
+ }  
+ tnp.addNegative(file, info.id);  
+ }  
+ public void onReadPrepareRecord(final long transactionID, final byte[] extraData, final int numberOfRecords) throws Exception  
+ {  
+ hasData.set(true);  
+ TransactionHolder tx = loadTransactions.get(transactionID);  
+ if (tx == null)  
+ {  
+ tx = new TransactionHolder(transactionID);  
+ loadTransactions.put(transactionID, tx);  
+ }  
+ tx.prepared = true;  
+ tx.extraData = extraData;  
+ JournalTransaction journalTransaction = transactions.get(transactionID);  
+ if (journalTransaction == null)  
+ {  
+ journalTransaction = new JournalTransaction(transactionID, JournalImpl.this);  
+ transactions.put(transactionID, journalTransaction);  
+ }  
+ boolean healthy = checkTransactionHealth(file, journalTransaction, orderedFiles, numberOfRecords);  
+ if (healthy)  
+ {  
+ journalTransaction.prepare(file);  
+ }  
+ else  
+ {  
+ HornetQJournalLogger.LOGGER.preparedTXIncomplete(transactionID);  
+ tx.invalid = true;  
+ }  
+ }  
+ public void onReadCommitRecord(final long transactionID, final int numberOfRecords) throws Exception  
+ {  
+ TransactionHolder tx = loadTransactions.remove(transactionID);  
+ if (tx != null)  
+ {  
+ JournalTransaction journalTransaction = transactions.remove(transactionID);  
+ if (journalTransaction == null)  
+ {  
+ throw new IllegalStateException("Cannot find tx " + transactionID);  
+ }  
+ boolean healthy = checkTransactionHealth(file, journalTransaction, orderedFiles, numberOfRecords);  
+ if (healthy)  
+ {  
+ for (RecordInfo txRecord : tx.recordInfos)  
+ {  
+ if (txRecord.isUpdate)  
+ {  
+ loadManager.updateRecord(txRecord);  
+ }  
+ else  
+ {  
+ loadManager.addRecord(txRecord);  
+ }  
+ }  
+ for (RecordInfo deleteValue : tx.recordsToDelete)  
+ {  
+ loadManager.deleteRecord(deleteValue.id);  
+ }  
+ journalTransaction.commit(file);  
+ }  
+ else  
+ {  
+ HornetQJournalLogger.LOGGER.txMissingElements(transactionID);  
+ journalTransaction.forget();  
+ }  
+ hasData.set(true);  
+ }  
+ }  
+ public void onReadRollbackRecord(final long transactionID) throws Exception  
+ {  
+ TransactionHolder tx = loadTransactions.remove(transactionID);  
+ if (tx != null)  
+ {  
+ JournalTransaction tnp = transactions.remove(transactionID);  
+ if (tnp == null)  
+ {  
+ throw new IllegalStateException("Cannot find tx " + transactionID);  
+ }  
+ tnp.rollback(file);  
+ hasData.set(true);  
+ }  
+ }  
+ public void markAsDataFile(final JournalFile file)  
+ {  
+ hasData.set(true);  
+ }  
+ });  
+ if (hasData.get())  
+ {  
+ lastDataPos = resultLastPost;  
+ filesRepository.addDataFileOnBottom(file);  
+ }  
+ else  
+ {  
+ if (changeData)  
+ {  
+ filesRepository.addFreeFile(file, false, false);  
+ }  
+ }  
+ }  
+ if (replicationSync == JournalState.SYNCING)  
+ {  
**+ assert filesRepository.getDataFiles().isEmpty();**+ setJournalState(JournalState.SYNCING);  
+ return new JournalLoadInformation(0, -1);  
+ }  
+ setUpCurrentFile(lastDataPos);  
+ setJournalState(JournalState.LOADED);  
+ for (TransactionHolder transaction : loadTransactions.values())  
+ {  
+ if ((!transaction.prepared || transaction.invalid) && replicationSync != JournalState.SYNCING\_UP\_TO\_DATE)  
+ {  
+ HornetQJournalLogger.LOGGER.uncomittedTxFound(transaction.transactionID);  
+ if (changeData)  
+ {  
+ this.appendRollbackRecord(transaction.transactionID, false);  
+ }  
+ loadManager.failedTransaction(transaction.transactionID,  
+ transaction.recordInfos,  
+ transaction.recordsToDelete);  
+ }  
+ else  
+ {  
+ for (RecordInfo info : transaction.recordInfos)  
+ {  
+ if (info.id > maxID.get())  
+ {  
+ maxID.set(info.id);  
+ }  
+ }  
+ PreparedTransactionInfo info = new PreparedTransactionInfo(transaction.transactionID, transaction.extraData);  
+ info.records.addAll(transaction.recordInfos);  
+ info.recordsToDelete.addAll(transaction.recordsToDelete);  
+ loadManager.addPreparedTransaction(info);  
+ }  
+ }  
+ checkReclaimStatus();  
+ return new JournalLoadInformation(records.size(), maxID.longValue());  
+ }  
========================================

ACTIVEMQ6-4 - Rename packages to ActiveMQ  
https://issues.apache.org/jira/browse/ACTIVEMQ6-4  
Repackage the modules, java source and maven poms to apache.activemq6  
  
Project: activemq-artemis.txt  
Commit: 23e8edd9791b5a2ac025c321f97a9dd2329bbeaa  
ClassName: org.apache.activemq6.core.client.impl.ServerLocatorImpl.java  
  
+ public ClientSessionFactory createSessionFactory() throws HornetQException  
+ {  
**+ assertOpen();**+ initialise();  
+ if (this.getNumInitialConnectors() == 0 && discoveryGroup != null)  
+ {  
+ long timeout = clusterConnection ? 0 : discoveryGroupConfiguration.getDiscoveryInitialWaitTimeout();  
+ boolean ok = discoveryGroup.waitForBroadcast(timeout);  
+ if (!ok)  
+ {  
+ throw HornetQClientMessageBundle.BUNDLE.connectionTimedOutInInitialBroadcast();  
+ }  
+ }  
+ ClientSessionFactoryInternal factory = null;  
+ synchronized (this)  
+ {  
+ boolean retry;  
+ int attempts = 0;  
+ do  
+ {  
+ retry = false;  
+ TransportConfiguration tc = selectConnector();  
+ if (tc == null)  
+ {  
+ throw HornetQClientMessageBundle.BUNDLE.noTCForSessionFactory();  
+ }  
+ try  
+ {  
+ factory = new ClientSessionFactoryImpl(this,  
+ tc,  
+ callTimeout,  
+ callFailoverTimeout,  
+ clientFailureCheckPeriod,  
+ connectionTTL,  
+ retryInterval,  
+ retryIntervalMultiplier,  
+ maxRetryInterval,  
+ reconnectAttempts,  
+ threadPool,  
+ scheduledThreadPool,  
+ incomingInterceptors,  
+ outgoingInterceptors);  
+ try  
+ {  
+ addToConnecting(factory);  
+ factory.connect(initialConnectAttempts, failoverOnInitialConnection);  
+ }  
+ finally  
+ {  
+ removeFromConnecting(factory);  
+ }  
+ }  
+ catch (HornetQException e)  
+ {  
+ factory.close();  
+ factory = null;  
+ if (e.getType() == HornetQExceptionType.NOT\_CONNECTED)  
+ {  
+ attempts++;  
+ synchronized (topologyArrayGuard)  
+ {  
+ if (topologyArray != null && attempts == topologyArray.length)  
+ {  
+ throw HornetQClientMessageBundle.BUNDLE.cannotConnectToServers();  
+ }  
+ if (topologyArray == null && attempts == this.getNumInitialConnectors())  
+ {  
+ throw HornetQClientMessageBundle.BUNDLE.cannotConnectToServers();  
+ }  
+ }  
+ retry = true;  
+ }  
+ else  
+ {  
+ throw e;  
+ }  
+ }  
+ }  
+ while (retry);  
+ final long timeout = System.currentTimeMillis() + callTimeout;  
+ while (!isClosed() && !receivedTopology && timeout > System.currentTimeMillis())  
+ {  
+ try  
+ {  
+ wait(1000);  
+ }  
+ catch (InterruptedException e)  
+ {  
+ throw new HornetQInterruptedException(e);  
+ }  
+ }  
+ final boolean hasTimedOut = timeout > System.currentTimeMillis();  
+ if (!hasTimedOut && !receivedTopology)  
+ {  
+ if (factory != null)  
+ factory.cleanup();  
+ throw HornetQClientMessageBundle.BUNDLE.connectionTimedOutOnReceiveTopology(discoveryGroup);  
+ }  
+ addFactory(factory);  
+ return factory;  
+ }  
+ }  
========================================

accepting contributions from Lahiru to fix AIRAVATA-20 and AIRAVATA-21  
  
Project: airavata.txt  
Commit: 8118f98dd87aa358939fd46b46bfeb434a19b2d1  
ClassName: userguide.loggingmodule.LoggingModule.java  
  
**+ public boolean canSupportAssertion(Assertion assertion) {**+ return true;  
+ }  
========================================

AIRAVATA 101 fix test case for messagebox  
  
Project: airavata.txt  
Commit: 54b4c795f6212b7ab5e4ef9b7105869ea406c857  
ClassName: org.apache.airavata.wsmg.msgbox.InMemoryMessageBoxServer.java  
  
+ private static File getAddressingMARFile() {  
+ File dir = new File(prefixBaseDirectory(Constants.TESTING\_REPOSITORY + "/modules"));  
+ File[] files = dir.listFiles(new AddressingFilter());  
**+ TestCase.assertTrue(files.length == 1);**+ File file = files[0];  
**+ TestCase.assertTrue(file.exists());**+ return file;  
+ }  
========================================

fixing 406 error for application job data retrieval  
  
Project: airavata.txt  
Commit: dcf50a4a32329748818ca93e99977572b04715c0  
ClassName: org.apache.airavata.integration.BaseCaseIT.java  
  
@@ -431,4 +431,4 @@ public void notify(EventDataRepository eventDataRepo, EventData eventData) {  
 log.info("Verifying application jobs ....");  
+ List<ApplicationJob> applicationJobs = airavataAPI.getProvenanceManager().getApplicationJobs(experimentId, null, null);  
**+ Assert.assertEquals(applicationJobs.size(), 1);** }  
========================================

AIRAVATA 101 fix test case for messagebox  
  
Project: airavata.txt  
Commit: 54b4c795f6212b7ab5e4ef9b7105869ea406c857  
ClassName: org.apache.airavata.wsmg.msgbox.InMemoryMessageBoxServer.java  
  
+ public static ServiceContext createAdressedEnabledClientSide(AxisService service) throws AxisFault {  
+ File file = getAddressingMARFile();  
**+ TestCase.assertTrue(file.exists());**+ ConfigurationContext configContext = ConfigurationContextFactory.createConfigurationContextFromFileSystem(  
+ prefixBaseDirectory("target/test-resources/integrationRepo"), null);  
+ AxisModule axisModule = DeploymentEngine.buildModule(file, configContext.getAxisConfiguration());  
+ configContext.getAxisConfiguration().addModule(axisModule);  
+ configContext.getAxisConfiguration().addService(service);  
+ ServiceGroupContext serviceGroupContext = configContext  
+ .createServiceGroupContext(service.getAxisServiceGroup());  
+ return serviceGroupContext.getServiceContext(service);  
+ }  
========================================

AIRAVATA 101 fix test case for messagebox  
  
Project: airavata.txt  
Commit: 54b4c795f6212b7ab5e4ef9b7105869ea406c857  
ClassName: org.apache.airavata.wsmg.msgbox.InMemoryMessageBoxServer.java  
  
+ public static ServiceContext createAdressedEnabledClientSide(AxisService service, String clientHome)  
+ throws AxisFault {  
+ File file = getAddressingMARFile();  
**+ TestCase.assertTrue(file.exists());**+ ConfigurationContext configContext = ConfigurationContextFactory.createConfigurationContextFromFileSystem(  
+ clientHome, null);  
+ AxisModule axisModule = DeploymentEngine.buildModule(file, configContext.getAxisConfiguration());  
+ configContext.getAxisConfiguration().addModule(axisModule);  
+ configContext.getAxisConfiguration().addService(service);  
+ ServiceGroupContext serviceGroupContext = configContext  
+ .createServiceGroupContext(service.getAxisServiceGroup());  
+ return serviceGroupContext.getServiceContext(service);  
+ }  
========================================

AIRAVATA 101 fix test case for messagebox  
  
Project: airavata.txt  
Commit: 54b4c795f6212b7ab5e4ef9b7105869ea406c857  
ClassName: org.apache.airavata.wsmg.msgbox.InMemoryMessageBoxServer.java  
  
+ public static ConfigurationContext createClientConfigurationContext() throws AxisFault {  
+ File file = getAddressingMARFile();  
**+ TestCase.assertTrue(file.exists());**+ ConfigurationContext configContext = ConfigurationContextFactory.createConfigurationContextFromFileSystem(  
+ prefixBaseDirectory(Constants.TESTING\_PATH + "/integrationRepo"),  
+ prefixBaseDirectory(Constants.TESTING\_PATH + "/integrationRepo/conf/axis2.xml"));  
+ AxisModule axisModule = DeploymentEngine.buildModule(file, configContext.getAxisConfiguration());  
+ configContext.getAxisConfiguration().addModule(axisModule);  
+ return configContext;  
+ }  
========================================

1.) SAAJ now uses DOM Level-3 methods (due to SAAJ 1.3 impl) - making sure we build it only when we'r building on JDK 1.5  
2.) Applying Dimuthu's patch from (https://issues.apache.org/jira/browse/AXIS2-1963)  
  
Project: axis2-java.txt  
Commit: b6dd3526b4a0000485a1592d0250f364ba9b49cb  
**ClassName: org.apache.axis2.policy.builders.MTOMAssertionBuilder.java**+ public QName[] getKnownElements()   
+ {  
**+ return new QName[] {new QName(MTOMAssertion.NS, MTOMAssertion.MTOM\_SERIALIZATION\_CONFIG\_LN)};**+ }  
========================================

**Added a mechamism where one can declear the policy assertions (in module.xml) that the system should not exposed**to outside.  
Implemented the serialization method of the Trust10.  
Minor fixed for policy hadling mechanism in the generated stubs.  
  
Project: axis2-java.txt  
Commit: 244aa4f43ff8119055aa57a36d41877e9626c506  
ClassName: org.apache.axis2.engine.AxisConfiguration.java  
  
**+ public void removeLocalPolicyAssertion(QName name) {  
+ this.localPolicyAssertions.remove(name);**+ }  
========================================

- Updated the SAMLTokenIssuer to issue bearer tokens  
- Updated the sts cert to the standard STS cert used for interop scenarios  
- Added a test to test the STS issuing a SAML bearer token authenticating the requester using a UsernameToken  
  
Project: axis2-java.txt  
Commit: 4fdc35993db4e0b13f0b68336db8862df0500fdd  
ClassName: org.apache.rahas.impl.SAMLTokenIssuer.java  
  
**@@ -365,5 +353,54 @@ private SAMLAssertion createHoKAssertion(SAMLTokenIssuerConfig config,** }  
**- return this.createAssertion(doc, encryptedKeyElem,   
+ return this.createAttributeAssertion(doc, encryptedKeyElem,** config, crypto, creationTime, expirationTime);  
 }  
========================================

(1) Modified InterfaceImplementationTemplate to use the new Neethi2 code  
(2) Added the processing of Policies defined in an import of a WSDL  
(3) Minor bug fixes  
  
Project: axis2-java.txt  
Commit: a7c6eda1a7f4ace1af780c5deee6a5aa9b6e69e2  
ClassName: org.apache.axis2.wsdl.codegen.extension.PolicyEvaluator.java  
  
@@ -157,3 +157,4 @@ private void processPolicies(Document document, Element rootElement,  
 **for (Iterator assertions = ((List) iterator.next()).iterator(); assertions.hasNext();) {  
- assertion = (Assertion) iterator.next();  
+ assertion = (Assertion) assertions.next();  
 targetNamesapce = assertion.getName().getNamespaceURI();**========================================

IMPORTANT: This change requires the latest AXIOM snapshots  
- Rahas can now issue SignedSAMLTokens :-)  
- Improved the rahas SAMLtokenIssuer configuration  
  
Project: axis2-java.txt  
Commit: 3c74807d67054ac4800e9678efe2f89f05bf6c6e  
ClassName: org.apache.rahas.impl.SAMLTokenIssuer.java  
  
@@ -176,3 +177,3 @@ public SOAPEnvelope issue(OMElement request, MessageContext inMsgCtx)  
**- SAMLAssertion assertion = this.createAssertion(doc, encryptedKeyElem, config);  
+ SAMLAssertion assertion = this.createAssertion(doc, encryptedKeyElem, config, crypto);**========================================

- Updated the SAMLTokenIssuer to issue bearer tokens  
- Updated the sts cert to the standard STS cert used for interop scenarios  
- Added a test to test the STS issuing a SAML bearer token authenticating the requester using a UsernameToken  
  
Project: axis2-java.txt  
Commit: 4fdc35993db4e0b13f0b68336db8862df0500fdd  
ClassName: org.apache.rahas.impl.SAMLTokenIssuer.java  
  
**- private SAMLAssertion createAssertion(Document doc,   
+ private SAMLAssertion createAttributeAssertion(Document doc,** Element keyInfoContent,   
========================================

- Updated the SAMLTokenIssuer to issue bearer tokens  
- Updated the sts cert to the standard STS cert used for interop scenarios  
- Added a test to test the STS issuing a SAML bearer token authenticating the requester using a UsernameToken  
  
Project: axis2-java.txt  
Commit: 4fdc35993db4e0b13f0b68336db8862df0500fdd  
ClassName: org.apache.rahas.impl.SAMLTokenIssuer.java  
  
**+ private SAMLAssertion createAuthAssertion(String confMethod,**+ SAMLNameIdentifier subjectNameId, SAMLTokenIssuerConfig config,  
+ Crypto crypto, Date notBefore, Date notAfter) throws TrustException {  
+ try {  
+ String[] confirmationMethods = new String[]{confMethod};  
+ SAMLSubject subject = new SAMLSubject(subjectNameId, Arrays.asList(confirmationMethods), null, null);  
+ SAMLAuthenticationStatement authStmt = new SAMLAuthenticationStatement(  
+ subject,  
+ SAMLAuthenticationStatement.AuthenticationMethod\_Password,  
+ notBefore, null, null, null);  
+ SAMLStatement[] statements = {authStmt};  
**+ SAMLAssertion assertion = new SAMLAssertion(config.issuerName, notBefore,**+ notAfter, null, null, Arrays.asList(statements));  
+ X509Certificate[] issuerCerts =  
+ crypto.getCertificates(config.issuerKeyAlias);  
+ String sigAlgo = XMLSignature.ALGO\_ID\_SIGNATURE\_RSA;  
+ String pubKeyAlgo =  
+ issuerCerts[0].getPublicKey().getAlgorithm();  
+ if (pubKeyAlgo.equalsIgnoreCase("DSA")) {  
+ sigAlgo = XMLSignature.ALGO\_ID\_SIGNATURE\_DSA;  
+ }  
+ java.security.Key issuerPK =  
+ crypto.getPrivateKey(config.issuerKeyAlias,  
+ config.issuerKeyPassword);  
**+ assertion.sign(sigAlgo, issuerPK, Arrays.asList(issuerCerts));  
+ return assertion;**+ } catch (Exception e) {  
**+ throw new TrustException("samlAssertionCreationError", e);**+ }  
+ }   
========================================

- Updated the SAMLTokenIssuer to issue a SAMLToken (HoK) with a AuthStatement containing the requester's cert by authenticating the requester with signature.  
- Added a testcase to test the above case  
- Fixed AXIS2-848  
  
Project: axis2-java.txt  
Commit: 5203840ede53c3b77f2e490250f6d0fbc377182d  
ClassName: org.apache.rahas.impl.SAMLTokenIssuer.java  
  
**- private SAMLAssertion createAuthAssertion(String confMethod,**- SAMLNameIdentifier subjectNameId, SAMLTokenIssuerConfig config,  
**+ private SAMLAssertion createAuthAssertion(Document doc, String confMethod,**+ SAMLNameIdentifier subjectNameId, Element keyInfoContent, SAMLTokenIssuerConfig config,  
 Crypto crypto, Date notBefore, Date notAfter) throws TrustException {  
========================================

**Added a mechamism where one can declear the policy assertions (in module.xml) that the system should not exposed**to outside.  
Implemented the serialization method of the Trust10.  
Minor fixed for policy hadling mechanism in the generated stubs.  
  
Project: axis2-java.txt  
Commit: 244aa4f43ff8119055aa57a36d41877e9626c506  
ClassName: org.apache.axis2.deployment.ModuleBuilder.java  
  
@@ -156,2 +156,14 @@ public void populateModule() throws DeploymentException {  
**+ OMElement localPolicyAssertionElement = moduleElement.getFirstChildWithName(new QName("local-policy-assertions"));  
+ if (localPolicyAssertionElement != null) {  
+ module.setLocalPolicyAssertions(getLocalPolicyAssertionNames(localPolicyAssertionElement));**+ }  
========================================

- Added the AppliesTo element to refer to the service the client is trying to talk to when requesting for a SAMLToken  
- Added the KeySize and KeyType elements to the requests of the two Rahas tests and no the SAMLTokenIssuer issues SAMLTokens based on the key type.  
  
Project: axis2-java.txt  
Commit: dec8c9a22a212941bb0179d66d90480b215e611a  
ClassName: org.apache.rahas.impl.SAMLTokenIssuer.java  
  
@@ -184,6 +184,20 @@ public SOAPEnvelope issue(OMElement request, MessageContext inMsgCtx)  
 String keyType = TrustUtil.findKeyType(request);  
+ String appliesToAddress = this.getServiceAddress(request);  
**+ SAMLAssertion assertion = null;**+ if(keyType == null) {  
+ throw new TrustException(TrustException.INVALID\_REQUEST, new String[]{"Requested KeyType is missing"});  
+ }  
========================================

Support pushing sorts down to MongoDB. Required new rule PushSortPastProjectRule, and some bug fixes in how Optiq deals with subsets. If a rel requires a rel sorted by [x] it should also allow one sorted by [x, y]. A few rules were incorrectly assigning the collation of their result.  
Make JoinRel.stoppedVariables an immutable set, and final.  
Add Util.isDistinct(), and use it in various places.  
Add instructions for loading FoodMart data set into MongoDB.  
  
Project: calcite.txt  
Commit: 8ad8e23b47650a669e80209eb1f93716a4b19857  
ClassName: net.hydromatic.optiq.util.PartiallyOrderedSet.java  
  
+ private Set<Node<E>> findParentsChildren(  
+ E e,  
+ ArrayDeque<Node<E>> ancestors,  
+ boolean up) {  
+ final Set<Node<E>> parents = new HashSet<Node<E>>();  
+ while (!ancestors.isEmpty()) {  
+ final Node<E> ancestor = ancestors.pop();  
**+ assert ancestor.e == null**+ || (up  
+ ? !ordering.lessThan(ancestor.e, e)  
+ : !ordering.lessThan(e, ancestor.e));  
**+ assert ancestor.e != e; // e not in tree yet**+ int found = 0;  
+ for (Node<E> child : (up ? ancestor.childList : ancestor.parentList)) {  
+ if (child.e == null) {  
+ continue; // child is the bottom node  
+ }  
+ if (up  
+ ? ordering.lessThan(e, child.e)  
+ : ordering.lessThan(child.e, e)) {  
+ ancestors.add(child);  
+ ++found;  
+ } else if (up  
+ ? !ordering.lessThan(child.e, e)  
+ : !ordering.lessThan(e, child.e)) {  
+ ancestors.add(child);  
+ }  
+ }  
+ if (found == 0) {  
+ if (ancestor.e == null  
+ || (up  
+ ? ordering.lessThan(e, ancestor.e)  
+ : ordering.lessThan(ancestor.e, e))) {  
+ parents.add(ancestor);  
+ }  
+ }  
+ }  
+ return parents;  
+ }  
========================================

Fix a few issues with collation as a trait.  
  
Project: calcite.txt  
Commit: 07e1cb0a6c30677502e63cb8e337f25c0a76e983  
ClassName: org.eigenbase.rel.ProjectRel.java  
  
 public RelNode copy(RelTraitSet traitSet, List<RelNode> inputs) {  
**- assert traitSet.comprises(Convention.NONE);  
+ assert traitSet.contains(Convention.NONE);** return new ProjectRel(  
========================================

Fix issues with timezone offsets.  
  
Project: calcite.txt  
Commit: 9c6b374f6f14cc9739d2ef55ced257333fa32643  
ClassName: net.hydromatic.optiq.runtime.AbstractCursor.java  
  
@@ -114,3 +119,3 @@ protected Accessor createAccessor(ColumnMetaData type, int ordinal) {  
 default:  
**- throw new AssertionError("bad " + type.internalClass);  
+ throw new AssertionError("bad " + type.representation);** }  
========================================

Fix bug in how ArrayTable automatically sorts records.  
  
Project: calcite.txt  
Commit: aeb9bfdd7587657b855ae23aee250836fa46fa3c  
ClassName: net.hydromatic.optiq.impl.clone.ArrayTable.java  
  
+ private static String toString(Primitive p, Object dataSet) {  
+ Bug.upgrade("use Primitive.arrayToString after linq4j upgrade");  
+ switch (p) {  
+ case BOOLEAN:  
+ return Arrays.toString((boolean[]) dataSet);  
+ case BYTE:  
+ return Arrays.toString((byte[]) dataSet);  
+ case CHAR:  
+ return Arrays.toString((char[]) dataSet);  
+ case DOUBLE:  
+ return Arrays.toString((double[]) dataSet);  
+ case FLOAT:  
+ return Arrays.toString((float[]) dataSet);  
+ case INT:  
+ return Arrays.toString((int[]) dataSet);  
+ case LONG:  
+ return Arrays.toString((long[]) dataSet);  
+ case OTHER:  
+ case VOID:  
+ return Arrays.toString((Object[]) dataSet);  
+ case SHORT:  
+ return Arrays.toString((short[]) dataSet);  
+ default:  
**+ throw new AssertionError(p);**+ }  
+ }  
========================================

Support pushing sorts down to MongoDB. Required new rule PushSortPastProjectRule, and some bug fixes in how Optiq deals with subsets. If a rel requires a rel sorted by [x] it should also allow one sorted by [x, y]. A few rules were incorrectly assigning the collation of their result.  
Make JoinRel.stoppedVariables an immutable set, and final.  
Add Util.isDistinct(), and use it in various places.  
Add instructions for loading FoodMart data set into MongoDB.  
  
Project: calcite.txt  
Commit: 8ad8e23b47650a669e80209eb1f93716a4b19857  
ClassName: org.eigenbase.relopt.volcano.VolcanoPlanner.java  
  
@@ -1166,4 +1167,4 @@ public int compare(  
 input = rels.next();  
**- assert inputSubset.getTraitSet().equals(**- input.getTraitSet());  
**+ assert input.getTraitSet().subsumes(**+ inputSubset.getTraitSet());  
 **assert inputSet.rels.contains(input);**========================================

More null-semantics fixes.  
The 'item' operator can now operate on an 'any' object. (Simulates dynamic typing, needed by Drill.)  
Add "enabled" flag to SqlOperatorBaseTest (set to false by OptiqSqlOperatorTest) to mark out which tests (or parts of tests) currently pass.  
  
Project: calcite.txt  
Commit: 7cac28b15c558ddbbd7c98b5099584e1deb57682  
ClassName: org.eigenbase.sql.fun.SqlItemOperator.java  
  
+ private SqlSingleOperandTypeChecker getChecker(RelDataType operandType) {  
 switch (operandType.getSqlTypeName()) {  
 case ARRAY:  
- checker = SqlTypeStrategies.family(SqlTypeFamily.INTEGER);  
- break;  
- default:  
+ return SqlTypeStrategies.family(SqlTypeFamily.INTEGER);  
 case MAP:  
- checker = SqlTypeStrategies.family(  
+ return SqlTypeStrategies.family(  
 operandType.getKeyType().getSqlTypeName().getFamily());  
- break;  
+ case ANY:  
+ return SqlTypeStrategies.or(  
+ SqlTypeStrategies.family(SqlTypeFamily.INTEGER),  
+ SqlTypeStrategies.family(SqlTypeFamily.CHARACTER));  
+ default:  
**+ throw new AssertionError(operandType.getSqlTypeName());** }  
========================================

Fix a few issues with collation as a trait.  
  
Project: calcite.txt  
Commit: 07e1cb0a6c30677502e63cb8e337f25c0a76e983  
ClassName: org.eigenbase.rel.FilterRel.java  
  
 public RelNode copy(RelTraitSet traitSet, List<RelNode> inputs) {  
**- assert traitSet.comprises(Convention.NONE);  
+ assert traitSet.contains(Convention.NONE);** return new FilterRel(  
========================================

Lots of fixes to expressions, especially CAST.  
  
Project: calcite.txt  
Commit: ce782d607b958ffa5d904277977f813eb5552cb5  
ClassName: org.eigenbase.rex.RexLiteral.java  
  
@@ -492,3 +496,3 @@ public Comparable getValue()  
 {  
**- assert valueMatchesType(value, typeName) : value;  
+ assert valueMatchesType(value, typeName, true) : value;** return value;  
========================================

Some satisfiability fixes.  
  
Project: calcite.txt  
Commit: bd8d871abf48510119e6f44c861062b686b12b7c  
ClassName: org.eigenbase.sql.SqlBinaryOperator.java  
  
+ public boolean validRexOperands(int count, boolean fail) {  
+ if (count != 2) {  
+ if ((this == SqlStdOperatorTable.andOperator  
+ || this == SqlStdOperatorTable.orOperator)  
+ && count > 2)  
+ {  
+ return true;  
+ }  
**+ assert !fail : "wrong operand count " + count + " for " + this;**+ return false;  
+ }  
========================================

Make RelCollation a trait. This will allow us to ask for relational expressions that are equivalent (i.e. in the same set) but have different sort order. And will allow us to apply a converter (i.e. an OrderRel) to create a rel with the desired sort order.  
RelCollation is the first trait where it is reasonable for a given RelNode to have 2 instances of the trait. For example, the TableAccessRel(table='TIME\_BY\_DAY') might be sorted by {the\_year, the\_month, the\_date} and also by {time\_id}. That particular TableAccessRel would belong to two subsets in the same set. That confuses the current RelSubset data structure, which assumes that a RelNode is in precisely one RelSubset. We've fixed it by moving the lists to RelSet. The lists of rels and parents in RelSubset are now computed on the fly.  
  
Project: calcite.txt  
Commit: a72e41a2af122b53fe302e195c190ab1adc952a2  
ClassName: org.eigenbase.relopt.volcano.VolcanoPlanner.java  
  
@@ -904,4 +888,6 @@ public int compare(  
 if (input instanceof RelSubset) {  
**- assert inputSubset.rels.size() > 0;**- input = inputSubset.rels.get(0);  
+ final Iterator<RelNode> rels =  
+ inputSubset.getRels().iterator();  
**+ assert rels.hasNext();**+ input = rels.next();  
 **assert inputSubset.getTraitSet().equals(**========================================

various fixes for the test cases to use direct: rather than queue: to remove some async test case errors  
  
Project: camel.txt  
Commit: 989bd17d958c850e1e31e4c0ba97409a0a5886fd  
ClassName: org.apache.camel.component.mock.MockEndpoint.java  
  
 **public void assertIsSatisfied() throws InterruptedException {  
- assertIsSatisfied(1000);  
+ assertIsSatisfied(sleepForEmptyTest);** }  
========================================

CAY-1966 SQLTemplate/SQLSelect positional parameter binding  
\* new org.apache.cayenne.velocity package  
\* generics fixes in unit tests  
  
Project: cayenne.txt  
Commit: c870954210bbe2838b49acd13ea3f4455c4e44fb  
ClassName: org.apache.cayenne.velocity.ResultDirectiveIT.java  
  
+ private Map<String, Object> selectForQuery(String sql) {  
+ SQLTemplate template = new SQLTemplate(Artist.class, sql);  
+ template.setColumnNamesCapitalization(CapsStrategy.UPPER);  
+ MockOperationObserver observer = new MockOperationObserver();  
+ runtime.getDataDomain().performQueries(  
+ Collections.singletonList(template),  
+ observer);  
+ List<Map<String, Object>> data = observer.rowsForQuery(template);  
**+ assertEquals(1, data.size());**+ Map<String, Object> row = data.get(0);  
+ return row;  
+ }  
========================================

CAY-1966 SQLTemplate/SQLSelect positional parameter binding  
\* new org.apache.cayenne.velocity package  
\* generics fixes in unit tests  
  
Project: cayenne.txt  
Commit: c870954210bbe2838b49acd13ea3f4455c4e44fb  
ClassName: org.apache.cayenne.velocity.ResultDirectiveIT.java  
  
+ private Map<String, Object> insertArtist() throws Exception {  
+ Map<String, Object> parameters = new HashMap<String, Object>();  
+ parameters.put("id", new Integer(1));  
+ parameters.put("name", "ArtistToTestResult");  
+ String templateString = "INSERT INTO ARTIST (ARTIST\_ID, ARTIST\_NAME, DATE\_OF\_BIRTH) "  
+ + "VALUES (#bind($id), #bind($name), #bind($dob))";  
+ SQLTemplate template = new SQLTemplate(Object.class, templateString);  
+ template.setParameters(parameters);  
+ DataNode node = new DataNode();  
+ node.setEntityResolver(runtime.getDataDomain().getEntityResolver());  
+ node.setRowReaderFactory(mock(RowReaderFactory.class));  
+ node.setAdapter(dbAdapter);  
+ SQLTemplateAction action = new SQLTemplateAction(template, node);  
+ Connection c = runtime  
+ .getDataDomain()  
+ .getDataNodes()  
+ .iterator()  
+ .next()  
+ .getDataSource()  
+ .getConnection();  
+ try {  
+ MockOperationObserver observer = new MockOperationObserver();  
+ action.performAction(c, observer);  
+ int[] batches = observer.countsForQuery(template);  
**+ assertNotNull(batches);  
+ assertEquals(1, batches.length);  
+ assertEquals(1, batches[0]);**+ }  
+ finally {  
+ c.close();  
+ }  
+ MockOperationObserver observer = new MockOperationObserver();  
+ SelectQuery query = new SelectQuery(Artist.class);  
+ runtime  
+ .getDataDomain()  
+ .performQueries(Collections.singletonList(query), observer);  
+ List<?> data = observer.rowsForQuery(query);  
**+ assertEquals(1, data.size());**+ DataRow row = (DataRow) data.get(0);  
+ return row;  
+ }  
========================================

better DB cleanup to prevent random test failures  
Specifically hoping to clear this Jenkins error:  
java.sql.SQLException: Attempt to insert null into a non-nullable column in statement [ALTER TABLE PAINTING ALTER COLUMN NEWCOL2 SET NOT NULL]  
at org.hsqldb.jdbc.Util.sqlException(Unknown Source)  
at org.hsqldb.jdbc.jdbcStatement.fetchResult(Unknown Source)  
at org.hsqldb.jdbc.jdbcStatement.execute(Unknown Source)  
at org.apache.cayenne.merge.AbstractToDbToken.executeSql(AbstractToDbToken.java:57)  
at org.apache.cayenne.merge.AbstractToDbToken.execute(AbstractToDbToken.java:45)  
at org.apache.cayenne.merge.MergeCase.execute(MergeCase.java:131)  
**at org.apache.cayenne.merge.MergeCase.assertTokensAndExecute(MergeCase.java:188)**at org.apache.cayenne.merge.SetAllowNullToDbTest.test(SetAllowNullToDbTest.java:43)  
  
Project: cayenne.txt  
Commit: 3e04cda7fa14c61f051d1713d08b6e4578d23368  
ClassName: org.apache.cayenne.merge.MergeCase.java  
  
**- protected void assertTokens(**- List<MergerToken> tokens,  
- int expectedToDb,  
**+ protected void assertTokens(List<MergerToken> tokens, int expectedToDb,** int expectedToModel) {  
========================================

partially fixing random ARTGROUP cleanup code..  
  
Project: cayenne.txt  
Commit: eefbc9264c869db5f04d073df908a5c640e1a1d8  
ClassName: org.apache.cayenne.merge.MergeCase.java  
  
**- protected void assertTokens(List<MergerToken> tokens, int expectedToDb,**- int expectedToModel) {  
**+ protected void assertTokens(List<MergerToken> tokens, int expectedToDb, int expectedToModel) {** int actualToDb = 0;  
========================================

testing on MySQL - fixing some unit tests that fail only cause BigDecimals retuedn by MySQL used different precision  
  
Project: cayenne.txt  
Commit: d1ee5c97cf7ad9057bd6ff4c1f033f3dec43f3a1  
ClassName: org.apache.cayenne.unit.BasicCase.java  
  
**+ public static void assertEquals(BigDecimal d1, Object d2, double delta) {  
+ assertTrue(d2 instanceof BigDecimal);**+ BigDecimal d3 = d1.subtract((BigDecimal) d2);  
**+ assertTrue(Math.abs(d3.doubleValue()) < delta);**+ }  
========================================

better DB cleanup to prevent random test failures  
Specifically hoping to clear this Jenkins error:  
java.sql.SQLException: Attempt to insert null into a non-nullable column in statement [ALTER TABLE PAINTING ALTER COLUMN NEWCOL2 SET NOT NULL]  
at org.hsqldb.jdbc.Util.sqlException(Unknown Source)  
at org.hsqldb.jdbc.jdbcStatement.fetchResult(Unknown Source)  
at org.hsqldb.jdbc.jdbcStatement.execute(Unknown Source)  
at org.apache.cayenne.merge.AbstractToDbToken.executeSql(AbstractToDbToken.java:57)  
at org.apache.cayenne.merge.AbstractToDbToken.execute(AbstractToDbToken.java:45)  
at org.apache.cayenne.merge.MergeCase.execute(MergeCase.java:131)  
**at org.apache.cayenne.merge.MergeCase.assertTokensAndExecute(MergeCase.java:188)**at org.apache.cayenne.merge.SetAllowNullToDbTest.test(SetAllowNullToDbTest.java:43)  
  
Project: cayenne.txt  
Commit: 3e04cda7fa14c61f051d1713d08b6e4578d23368  
ClassName: org.apache.cayenne.merge.MergeCase.java  
  
**- protected void assertTokensAndExecute(**- DataNode node,  
- DataMap map,  
- int expectedToDb,  
- int expectedToModel) {  
**+ protected void assertTokensAndExecute(DataNode node, DataMap map,**+ int expectedToDb, int expectedToModel) {  
 List<MergerToken> tokens = createMergeTokens();  
========================================

InMemoryProvider:  
bugfix versioning: only checked out documents can be modified for versioned documents  
add test to update older versions  
clean up testing:  
- allow tests to use direct implementation of the services or go through client provider layer  
- remove all dependencies to client provider from other locations than test initialization  
put commonly used functionality for tests and test initialization in a common superclass  
Several fixes in Versioning  
Refactoring configuration parameters  
Documentation  
  
Project: chemistry.txt  
Commit: d01e33de930eda0ff584ac1acf40025774f6da41  
ClassName: org.apache.opencmis.inmemory.AbstractServiceTst.java  
  
+ protected ObjectData getDocumentObjectData(String id) {  
+ ObjectData res = null;  
+ try {  
+ String returnedId=null;  
+ res = fObjSvc.getObject(fRepositoryId, id, "\*", false, IncludeRelationships.NONE,  
+ null, false, false, null);  
**+ assertNotNull(res);**+ returnedId = res.getId();  
+ testReturnedProperties(returnedId, res.getProperties().getProperties());  
**+ assertEquals(id, returnedId);**+ } catch (Exception e) {  
+ fail("getObject() failed with exception: " + e);  
+ }   
+ return res;  
+ }  
========================================

InMemoryProvider:  
bugfix versioning: only checked out documents can be modified for versioned documents  
add test to update older versions  
clean up testing:  
- allow tests to use direct implementation of the services or go through client provider layer  
- remove all dependencies to client provider from other locations than test initialization  
put commonly used functionality for tests and test initialization in a common superclass  
Several fixes in Versioning  
Refactoring configuration parameters  
Documentation  
  
Project: chemistry.txt  
Commit: d01e33de930eda0ff584ac1acf40025774f6da41  
ClassName: org.apache.opencmis.inmemory.AbstractServiceTst.java  
  
+ private void initializeUsingClientProvider(Map<String, String> parameters) {  
+ CmisProvider provider;  
+ CmisProviderFactory factory = CmisProviderFactory.newInstance();  
+ provider = factory.createCmisProvider(parameters);  
**+ assertNotNull(provider);**+ fFactory = provider.getObjectFactory();  
+ fRepSvc = provider.getRepositoryService();  
+ fObjSvc = provider.getObjectService();  
+ fNavSvc = provider.getNavigationService();  
+ }  
========================================

InMemoryProvider:  
bugfix versioning: only checked out documents can be modified for versioned documents  
add test to update older versions  
clean up testing:  
- allow tests to use direct implementation of the services or go through client provider layer  
- remove all dependencies to client provider from other locations than test initialization  
put commonly used functionality for tests and test initialization in a common superclass  
Several fixes in Versioning  
Refactoring configuration parameters  
Documentation  
  
Project: chemistry.txt  
Commit: d01e33de930eda0ff584ac1acf40025774f6da41  
ClassName: org.apache.opencmis.inmemory.AbstractServiceTst.java  
  
+ protected String getPathOfFolder(String id) {   
+ String path=null;  
+ try {  
+ String filter = PropertyIds.CMIS\_PATH;  
+ PropertiesData res = fObjSvc.getProperties(fRepositoryId, id, filter, null);  
**+ assertNotNull(res);**+ PropertyData<String> pd = (PropertyData<String>) res.getProperties().get(PropertyIds.CMIS\_PATH);  
**+ assertNotNull(pd);**+ path = pd.getFirstValue();  
**+ assertNotNull(path);**+ } catch (Exception e) {  
+ fail("getProperties() failed with exception: " + e);  
+ }   
+ return path;  
+ }  
========================================

InMemoryProvider:  
bugfix versioning: only checked out documents can be modified for versioned documents  
add test to update older versions  
clean up testing:  
- allow tests to use direct implementation of the services or go through client provider layer  
- remove all dependencies to client provider from other locations than test initialization  
put commonly used functionality for tests and test initialization in a common superclass  
Several fixes in Versioning  
Refactoring configuration parameters  
Documentation  
  
Project: chemistry.txt  
Commit: d01e33de930eda0ff584ac1acf40025774f6da41  
ClassName: org.apache.opencmis.inmemory.AbstractServiceTst.java  
  
+ protected String getDocument(String id) {  
+ ObjectData res = getDocumentObjectData(id);  
**+ assertNotNull(res);**+ return res.getId();  
+ }  
========================================

InMemoryProvider:  
bugfix versioning: only checked out documents can be modified for versioned documents  
add test to update older versions  
clean up testing:  
- allow tests to use direct implementation of the services or go through client provider layer  
- remove all dependencies to client provider from other locations than test initialization  
put commonly used functionality for tests and test initialization in a common superclass  
Several fixes in Versioning  
Refactoring configuration parameters  
Documentation  
  
Project: chemistry.txt  
Commit: d01e33de930eda0ff584ac1acf40025774f6da41  
ClassName: org.apache.opencmis.inmemory.AbstractServiceTst.java  
  
+ protected void verifyContentResult(ContentStreamData sd) {  
**+ assertEquals("text/plain", sd.getMimeType());  
+ assertEquals("data.txt", sd.getFilename());  
+ assertEquals(32 \* 1024, sd.getLength().longValue());**+ byte[] ba = new byte[32];  
+ InputStream is = sd.getStream();  
+ int counter = 0;  
+ try {  
+ while (is.read(ba) == ba.length) {  
+ ++counter;   
**+ assertEquals(0x61, ba[0]);  
+ assertEquals(0x6e, ba[29]);  
+ assertEquals(0x0c, ba[30]);  
+ assertEquals(0x0a, ba[31]);**+ }  
+ }  
+ catch (IOException e) {  
+ fail("reading from content stream failed");  
+ }  
**+ assertEquals(1024, counter);**+ }  
========================================

InMemoryProvider:  
bugfix versioning: only checked out documents can be modified for versioned documents  
add test to update older versions  
clean up testing:  
- allow tests to use direct implementation of the services or go through client provider layer  
- remove all dependencies to client provider from other locations than test initialization  
put commonly used functionality for tests and test initialization in a common superclass  
Several fixes in Versioning  
Refactoring configuration parameters  
Documentation  
  
Project: chemistry.txt  
Commit: d01e33de930eda0ff584ac1acf40025774f6da41  
ClassName: org.apache.opencmis.inmemory.AbstractServiceTst.java  
  
+ protected void setUp(String typeCreatorClassName) throws Exception {  
  
+ LOG.debug("Initializing InMemory Test with type creator class: " + typeCreatorClassName);  
  
+ Map<String, String> parameters = new HashMap<String, String>();  
+ parameters.put(SessionParameter.BINDING\_SPI\_CLASS, CmisProviderFactory.BINDING\_SPI\_INMEMORY);  
+ parameters.put(ConfigConstants.TYPE\_CREATOR\_CLASS, typeCreatorClassName);  
  
+ parameters.put(ConfigConstants.REPOSITORY\_ID, REPOSITORY\_ID);  
+ parameters  
+ .put(ConfigConstants.REPOSITORY\_INFO\_CREATOR\_CLASS, UnitTestRepositoryInfo.class.getName());  
+ addParameters(parameters);  
+ if (fUseClientProviderInterface)  
+ initializeUsingClientProvider(parameters);  
+ else  
+ initializeDirect(parameters);  
**+ assertNotNull(fRepSvc);  
+ assertNotNull(fObjSvc);  
+ assertNotNull(fNavSvc);**+ RepositoryInfoData rep = fRepSvc.getRepositoryInfo(REPOSITORY\_ID, null);  
+ fRootFolderId = rep.getRootFolderId();  
+ fRepositoryId = rep.getRepositoryId();  
**+ assertNotNull(fRepositoryId);  
+ assertNotNull(fRootFolderId);**+ RuntimeContext.getRuntimeConfig().attachCfg(new DummyCallContext());  
+ }  
========================================

InMemoryProvider:  
bugfix versioning: only checked out documents can be modified for versioned documents  
add test to update older versions  
clean up testing:  
- allow tests to use direct implementation of the services or go through client provider layer  
- remove all dependencies to client provider from other locations than test initialization  
put commonly used functionality for tests and test initialization in a common superclass  
Several fixes in Versioning  
Refactoring configuration parameters  
Documentation  
  
Project: chemistry.txt  
Commit: d01e33de930eda0ff584ac1acf40025774f6da41  
ClassName: org.apache.opencmis.inmemory.AbstractServiceTst.java  
  
+ protected String getPathOfDocument(String id) {   
+ String path=null;  
+ String filter = "\*";   
+ List<ObjectParentData> parentData = fNavSvc.getObjectParents(fRepositoryId, id, filter, false, IncludeRelationships.NONE, null, true, null);  
+ String name = parentData.get(0).getRelativePathSegment();  
+ PropertyData<String> pd = (PropertyData<String>) parentData.get(0).getObject().getProperties().getProperties().get(PropertyIds.CMIS\_PATH);  
**+ assertNotNull(pd);**+ path = pd.getFirstValue() + "/" + name;  
+ return path;  
+ }  
========================================

InMemoryProvider:  
bugfix versioning: only checked out documents can be modified for versioned documents  
add test to update older versions  
clean up testing:  
- allow tests to use direct implementation of the services or go through client provider layer  
- remove all dependencies to client provider from other locations than test initialization  
put commonly used functionality for tests and test initialization in a common superclass  
Several fixes in Versioning  
Refactoring configuration parameters  
Documentation  
  
Project: chemistry.txt  
Commit: d01e33de930eda0ff584ac1acf40025774f6da41  
ClassName: org.apache.opencmis.inmemory.AbstractServiceTst.java  
  
+ protected String getByPath(String id, String path) {  
+ ObjectData res = null;  
+ try {  
+ res = fObjSvc.getObjectByPath(fRepositoryId, path, "\*", false, IncludeRelationships.NONE,  
+ null, false, false, null);  
**+ assertNotNull(res);  
+ assertEquals(id, res.getId());**+ } catch (Exception e) {  
+ fail("getObject() failed with exception: " + e);  
+ }   
+ return res.getId();  
+ }  
========================================

InMemoryProvider:  
bugfix versioning: only checked out documents can be modified for versioned documents  
add test to update older versions  
clean up testing:  
- allow tests to use direct implementation of the services or go through client provider layer  
- remove all dependencies to client provider from other locations than test initialization  
put commonly used functionality for tests and test initialization in a common superclass  
Several fixes in Versioning  
Refactoring configuration parameters  
Documentation  
  
Project: chemistry-opencmis.txt  
Commit: 4d4de032edb53c61748e788324fd2d93c86d8430  
ClassName: org.apache.opencmis.inmemory.AbstractServiceTst.java  
  
+ protected void setUp(String typeCreatorClassName) throws Exception {  
  
+ LOG.debug("Initializing InMemory Test with type creator class: " + typeCreatorClassName);  
  
+ Map<String, String> parameters = new HashMap<String, String>();  
+ parameters.put(SessionParameter.BINDING\_SPI\_CLASS, CmisProviderFactory.BINDING\_SPI\_INMEMORY);  
+ parameters.put(ConfigConstants.TYPE\_CREATOR\_CLASS, typeCreatorClassName);  
  
+ parameters.put(ConfigConstants.REPOSITORY\_ID, REPOSITORY\_ID);  
+ parameters  
+ .put(ConfigConstants.REPOSITORY\_INFO\_CREATOR\_CLASS, UnitTestRepositoryInfo.class.getName());  
+ addParameters(parameters);  
+ if (fUseClientProviderInterface)  
+ initializeUsingClientProvider(parameters);  
+ else  
+ initializeDirect(parameters);  
**+ assertNotNull(fRepSvc);  
+ assertNotNull(fObjSvc);  
+ assertNotNull(fNavSvc);**+ RepositoryInfoData rep = fRepSvc.getRepositoryInfo(REPOSITORY\_ID, null);  
+ fRootFolderId = rep.getRootFolderId();  
+ fRepositoryId = rep.getRepositoryId();  
**+ assertNotNull(fRepositoryId);  
+ assertNotNull(fRootFolderId);**+ RuntimeContext.getRuntimeConfig().attachCfg(new DummyCallContext());  
+ }  
========================================

CMIS-254: fixed AtomPub client  
  
Project: chemistry-opencmis.txt  
Commit: 0dd47902c0ca28e73cd8250b6f5d1a0e1f4f6b25  
ClassName: org.apache.chemistry.opencmis.fit.runtime.AbstractWriteObjectIT.java  
  
+ public void createHugeDocument() throws IOException {  
+ ObjectId parentId = this.session.createObjectId(this.fixture.getTestRootId());  
+ String folderName = UUID.randomUUID().toString();  
+ String typeId = FixtureData.DOCUMENT\_TYPE\_ID.value();  
+ Map<String, Object> properties = new HashMap<String, Object>();  
+ properties.put(PropertyIds.NAME, folderName);  
+ properties.put(PropertyIds.OBJECT\_TYPE\_ID, typeId);  
+ String filename = UUID.randomUUID().toString();  
+ String mimetype = "application/octet-stream";  
+ ObjectId id = this.session.createDocument(properties, parentId, null, VersioningState.NONE);  
**+ assertNotNull(id);**+ Document doc = (Document) this.session.getObject(id);  
**+ assertNotNull(doc);**+ final int size = 1 \* 1024 \* 1024 \* 1024; // 2GB  
+ InputStream in = new InputStream() {  
+ private int counter = -1;  
+ @Override  
+ public int read() throws IOException {  
+ counter++;  
+ if (counter >= size) {  
+ return -1;  
+ }  
+ return '0' + (counter / 10);  
+ }  
+ };  
========================================

InMemoryProvider:  
bugfix versioning: only checked out documents can be modified for versioned documents  
add test to update older versions  
clean up testing:  
- allow tests to use direct implementation of the services or go through client provider layer  
- remove all dependencies to client provider from other locations than test initialization  
put commonly used functionality for tests and test initialization in a common superclass  
Several fixes in Versioning  
Refactoring configuration parameters  
Documentation  
  
Project: chemistry-opencmis.txt  
Commit: 4d4de032edb53c61748e788324fd2d93c86d8430  
ClassName: org.apache.opencmis.inmemory.AbstractServiceTst.java  
  
+ protected String getByPath(String id, String path) {  
+ ObjectData res = null;  
+ try {  
+ res = fObjSvc.getObjectByPath(fRepositoryId, path, "\*", false, IncludeRelationships.NONE,  
+ null, false, false, null);  
**+ assertNotNull(res);  
+ assertEquals(id, res.getId());**+ } catch (Exception e) {  
+ fail("getObject() failed with exception: " + e);  
+ }   
+ return res.getId();  
+ }  
========================================

Patch:  
https://issues.apache.org/jira/browse/CMIS-200  
  
Project: chemistry-opencmis.txt  
Commit: 09f24d473fbcdf09cf55aa63111a4867bee6d7a0  
ClassName: org.apache.chemistry.opencmis.fit.runtime.ReadOnlyTypeIT.java  
  
+ public void readTypeChildrenDocumentSkip() {  
+ ObjectType otd = this.session.getTypeDefinition(ObjectType.DOCUMENT\_BASETYPE\_ID);  
**+ Assert.assertNotNull(otd);**+ this.session.getDefaultContext().setMaxItemsPerPage(2);  
+ PagingIterable<ObjectType> pc = this.session.getTypeChildren(otd.getId(), true);  
**+ Assert.assertNotNull(pc);**+ PagingIterable<ObjectType> pcc = pc.skipTo(2).getPage(2);   
+ for (ObjectType ot1 : pcc) {  
+ ObjectType ot2 = this.session.getTypeDefinition(ot1.getId());  
**+ Assert.assertEquals(ot1.getId(), ot2.getId());**+ }  
========================================

Patch:  
https://issues.apache.org/jira/browse/CMIS-200  
  
Project: chemistry-opencmis.txt  
Commit: 09f24d473fbcdf09cf55aa63111a4867bee6d7a0  
ClassName: org.apache.chemistry.opencmis.fit.runtime.ReadOnlyTypeIT.java  
  
+ public void readTypeChildrenFolderSkip() {  
+ ObjectType otd = this.session.getTypeDefinition(ObjectType.FOLDER\_BASETYPE\_ID);  
**+ Assert.assertNotNull(otd);**+ this.session.getDefaultContext().setMaxItemsPerPage(2);  
+ PagingIterable<ObjectType> pc = this.session.getTypeChildren(otd.getId(), true);  
**+ Assert.assertNotNull(pc);**+ PagingIterable<ObjectType> pcc = pc.skipTo(0).getPage(2);   
+ for (ObjectType ot1 : pcc) {  
+ ObjectType ot2 = this.session.getTypeDefinition(ot1.getId());  
**+ Assert.assertEquals(ot1, ot2);**+ }  
========================================

Patch:  
https://issues.apache.org/jira/browse/CMIS-200  
  
Project: chemistry-opencmis.txt  
Commit: 09f24d473fbcdf09cf55aa63111a4867bee6d7a0  
ClassName: org.apache.chemistry.opencmis.fit.runtime.WriteObjectVersionIT.java  
  
@@ -53,3 +59,21 @@ public void checkOutDocs() {  
 **assertNotNull(d);  
- assertEquals(id, d.getId());  
+ assertEquals(this.checkdOutId, d.getId());**+ break; // check only first and only loop entry  
+ }  
========================================

InMemoryProvider:  
bugfix versioning: only checked out documents can be modified for versioned documents  
add test to update older versions  
clean up testing:  
- allow tests to use direct implementation of the services or go through client provider layer  
- remove all dependencies to client provider from other locations than test initialization  
put commonly used functionality for tests and test initialization in a common superclass  
Several fixes in Versioning  
Refactoring configuration parameters  
Documentation  
  
Project: chemistry-opencmis.txt  
Commit: 4d4de032edb53c61748e788324fd2d93c86d8430  
ClassName: org.apache.opencmis.inmemory.AbstractServiceTst.java  
  
+ protected ObjectData getDocumentObjectData(String id) {  
+ ObjectData res = null;  
+ try {  
+ String returnedId=null;  
+ res = fObjSvc.getObject(fRepositoryId, id, "\*", false, IncludeRelationships.NONE,  
+ null, false, false, null);  
**+ assertNotNull(res);**+ returnedId = res.getId();  
+ testReturnedProperties(returnedId, res.getProperties().getProperties());  
**+ assertEquals(id, returnedId);**+ } catch (Exception e) {  
+ fail("getObject() failed with exception: " + e);  
+ }   
+ return res;  
+ }  
========================================

Patch:  
https://issues.apache.org/jira/browse/CMIS-200  
  
Project: chemistry-opencmis.txt  
Commit: 09f24d473fbcdf09cf55aa63111a4867bee6d7a0  
ClassName: org.apache.chemistry.opencmis.fit.runtime.ReadOnlyDiscoverIT.java  
  
@@ -42,3 +42,23 @@ public void query() {  
 default:  
- PagingIterable<QueryResult> resultSet = this.session.query(FixtureData.QUERY.toString(), false, 2);  
+ PagingIterable<QueryResult> resultSet = this.session.query(FixtureData.QUERY.toString(), false);  
**+ Assert.assertNotNull(resultSet);**+ for (QueryResult o : resultSet) {  
**+ Assert.assertNotNull(o);**+ }  
+ break;  
+ }  
========================================

InMemoryProvider:  
bugfix versioning: only checked out documents can be modified for versioned documents  
add test to update older versions  
clean up testing:  
- allow tests to use direct implementation of the services or go through client provider layer  
- remove all dependencies to client provider from other locations than test initialization  
put commonly used functionality for tests and test initialization in a common superclass  
Several fixes in Versioning  
Refactoring configuration parameters  
Documentation  
  
Project: chemistry-opencmis.txt  
Commit: 4d4de032edb53c61748e788324fd2d93c86d8430  
ClassName: org.apache.opencmis.inmemory.AbstractServiceTst.java  
  
+ protected String getDocument(String id) {  
+ ObjectData res = getDocumentObjectData(id);  
**+ assertNotNull(res);**+ return res.getId();  
+ }  
========================================

CMIS-264: Fixed SharePoint 2010 document content issue  
  
Project: chemistry-opencmis.txt  
Commit: 7ef9e06869f5dd14c4c72596e9ccdf2645197f9a  
ClassName: org.apache.chemistry.opencmis.client.bindings.spi.atompub.Base64.java  
  
+ public static String encodeBytes(byte[] source, int off, int len) {  
+ String encoded = null;  
+ try {  
+ encoded = encodeBytes(source, off, len, NO\_OPTIONS);  
+ } catch (java.io.IOException ex) {  
**+ assert false : ex.getMessage();**+ } // end catch  
**+ assert encoded != null;**+ return encoded;  
+ } // end encodeBytes  
========================================

HADOOP-4709. Add several new features and bug fixes to Chukwa. Contributed by Jerome Boulon, Eric Yang, Andy Konwinski, and Ariel Rabkin.  
  
Project: chukwa.txt  
Commit: 2edc33ac59e6018766896b84849b6d31a86fe431  
ClassName: org.apache.hadoop.chukwa.datacollection.agent.MemLimitQueue.java  
  
- public void add(Chunk event) throws InterruptedException  
+ public void add(Chunk chunk) throws InterruptedException  
 {  
**- assert event != null: "can't enqueue null chunks";  
+ assert chunk != null: "can't enqueue null chunks";** synchronized(this) {  
- while(event.getData().length + dataSize > MAX\_MEM\_USAGE)  
+ while(chunk.getData().length + dataSize > MAX\_MEM\_USAGE)  
+ {  
+ try   
+ {   
 this.wait();  
- dataSize += event.getData().length;  
- queue.add(event);  
+ log.info("MemLimitQueue is full [" + dataSize +"]");  
+ }  
+ catch(InterruptedException e) {}  
+ }  
========================================

bug 10538: avoid duplicated config values insertion when multiple management servers start at the same time  
status 10538: resolved fixed  
  
Project: cloudstack.txt  
Commit: 9b54b8761a92b902e38de910d47a6cd1ecbf0296  
ClassName: com.cloud.storage.dao.DiskOfferingDaoImpl.java  
  
+ public DiskOfferingVO persistDeafultDiskOffering(DiskOfferingVO offering) {  
**+ assert offering.getUniqueName() != null : "unique name shouldn't be null for the disk offering";**+ DiskOfferingVO vo = findByUniqueName(offering.getUniqueName());  
+ if (vo != null) {  
+ return vo;  
+ }  
+ try {  
+ return persist(offering);  
+ } catch (EntityExistsException e) {  
+ return findByUniqueName(offering.getUniqueName());  
+ }  
+ }  
========================================

bug 11817: NAAS external network device support  
-made Netscaler, SRX, F5 network elements as pluggable service  
-added abstract load balancer device manager ExternaLoadBalancerDeviceManager  
-made both F5 and Netscaler pluggable service to extend ExternaLoadBalancerDeviceManager  
-added abstract firewall device manager ExternalFirewallDeviceManager  
-made SRX pluugable service to extende ExternalFirewallDeviceManager  
-added API's to configure and manage netscaler devices  
  
Project: cloudstack.txt  
Commit: 0b05badaaa3eafe249291e385b6b2fd7965d10da  
ClassName: com.cloud.network.ExternalLoadBalancerDeviceManagerImpl.java  
  
+ public boolean manageGuestNetworkWithExternalLoadBalancer(boolean add, Network guestConfig) throws ResourceUnavailableException, InsufficientCapacityException {  
+ if (guestConfig.getTrafficType() != TrafficType.Guest) {  
+ s\_logger.trace("External load balancer can only be user for guest networks.");  
+ return false;  
+ }  
+ long zoneId = guestConfig.getDataCenterId();  
+ DataCenterVO zone = \_dcDao.findById(zoneId);  
+ HostVO externalLoadBalancer = null;  
+ String provider = \_ntwkSrvcProviderDao.getProviderForServiceInNetwork(guestConfig.getId(), Service.Lb);  
+ long physicalNetworkId = guestConfig.getPhysicalNetworkId();  
+ if (add) {  
+ boolean retry = true;  
+ boolean provisionLB = false;  
+ while (retry) {  
+ GlobalLock deviceMapLock = GlobalLock.getInternLock("LoadBalancerAllocLock");  
+ Transaction txn = Transaction.currentTxn();  
+ try {  
+ if (deviceMapLock.lock(120)) {  
+ try {  
+ NetworkOfferingVO offering = \_networkOfferingDao.findById(guestConfig.getNetworkOfferingId());  
+ long lbDeviceId;  
+ txn.start();  
+ boolean dedicatedLB = offering.getDedicatedLB();  
+ try {  
+ ExternalLoadBalancerDeviceVO lbDevice = findSuitableLoadBalancerForNetwork(guestConfig, dedicatedLB);  
+ lbDeviceId = lbDevice.getId();  
+ NetworkExternalLoadBalancerVO networkLB = new NetworkExternalLoadBalancerVO(guestConfig.getId(), lbDeviceId);  
+ \_networkExternalLBDao.persist(networkLB);  
+ lbDevice.setAllocationState(dedicatedLB ? LBDeviceAllocationState.Dedicated : LBDeviceAllocationState.Shared);  
+ \_externalLoadBalancerDeviceDao.update(lbDeviceId, lbDevice);  
+ externalLoadBalancer = \_hostDao.findById(lbDevice.getHostId());  
+ txn.commit();  
+ provisionLB = false;  
+ retry = false;  
+ } catch (InsufficientCapacityException exception) {  
+ if (provisionLB) {  
+ retry = false;  
+ throw exception; // if already attempted once throw out of capacity exception  
+ }  
+ provisionLB = true; // if possible provision a LB appliance in the physical network  
+ }  
+ } finally {  
+ deviceMapLock.unlock();  
+ if (externalLoadBalancer == null) {  
+ txn.rollback();  
+ }  
+ }  
+ }  
+ } finally {  
+ deviceMapLock.releaseRef();  
+ }  
+ if (provisionLB) {  
+ List<ExternalLoadBalancerDeviceVO> providerLbDevices = \_externalLoadBalancerDeviceDao.listByProviderAndDeviceAllocationState(physicalNetworkId, provider, LBDeviceAllocationState.Provider);  
+ if ((providerLbDevices != null) && (!providerLbDevices.isEmpty())) {  
+ for (ExternalLoadBalancerDeviceVO lbProviderDevice : providerLbDevices) {  
+ if (lbProviderDevice.getState() == LBDeviceState.Enabled) {  
+ String lbIP = null;  
+ String username = null;  
+ String password = null;  
+ String publiInterface = null;  
+ String privateInterface = null;  
+ CreateLBApplianceCommand lbProvisionCmd = new CreateLBApplianceCommand(lbIP);  
+ Answer answer = \_agentMgr.easySend(lbProviderDevice.getHostId(), lbProvisionCmd);  
+ if (answer == null || !answer.getResult()) {  
+ s\_logger.trace("Could not provision load balancer instance on the load balancer device " + lbProviderDevice.getId());  
+ continue;  
+ }  
+ }  
+ }  
+ }  
+ }  
+ }  
+ } else {  
+ ExternalLoadBalancerDeviceVO lbDeviceVO = getExternalLoadBalancerForNetwork(guestConfig);   
**+ assert(lbDeviceVO != null) : "There is no device assigned to this network how did shutdown network ended up here??";**+ externalLoadBalancer = \_hostDao.findById(lbDeviceVO.getHostId());  
**+ assert (externalLoadBalancer != null) : "There is no device assigned to this network how did shutdown network ended up here??";**+ }  
+ long guestVlanTag = Long.parseLong(guestConfig.getBroadcastUri().getHost());  
+ String selfIp = NetUtils.long2Ip(NetUtils.ip2Long(guestConfig.getGateway()) + 1);  
+ String guestVlanNetmask = NetUtils.cidr2Netmask(guestConfig.getCidr());  
+ Integer networkRate = \_networkMgr.getNetworkRate(guestConfig.getId(), null);  
+ IpAddressTO ip = new IpAddressTO(guestConfig.getAccountId(), null, add, false, true, String.valueOf(guestVlanTag), selfIp, guestVlanNetmask, null, null, networkRate, false);  
+ IpAddressTO[] ips = new IpAddressTO[1];  
+ ips[0] = ip;  
+ IpAssocCommand cmd = new IpAssocCommand(ips);  
+ Answer answer = \_agentMgr.easySend(externalLoadBalancer.getId(), cmd);  
+ if (answer == null || !answer.getResult()) {  
+ String action = add ? "implement" : "shutdown";  
+ String answerDetails = (answer != null) ? answer.getDetails() : "answer was null";  
+ String msg = "External load balancer was unable to " + action + " the guest network on the external load balancer in zone " + zone.getName() + " due to " + answerDetails;  
+ s\_logger.error(msg);  
+ throw new ResourceUnavailableException(msg, DataCenter.class, zoneId);  
+ }  
+ List<String> reservedIpAddressesForGuestNetwork = \_nicDao.listIpAddressInNetwork(guestConfig.getId());  
+ if (add && (!reservedIpAddressesForGuestNetwork.contains(selfIp))) {  
+ savePlaceholderNic(guestConfig, selfIp);  
+ }  
+ if (!add) {  
+ Transaction txn = Transaction.currentTxn();  
+ txn.start();  
+ NetworkExternalLoadBalancerVO networkLBDevice = \_networkExternalLBDao.findByNetworkId(guestConfig.getId());  
+ \_networkExternalLBDao.remove(networkLBDevice.getId());  
+ List<NetworkExternalLoadBalancerVO> ntwksMapped = \_networkExternalLBDao.listByLoadBalancerDeviceId(networkLBDevice.getExternalLBDeviceId());  
+ if (ntwksMapped == null || ntwksMapped.isEmpty()) {  
+ ExternalLoadBalancerDeviceVO lbDevice = \_externalLoadBalancerDeviceDao.findById(networkLBDevice.getExternalLBDeviceId());  
+ lbDevice.setAllocationState(LBDeviceAllocationState.Free);  
+ \_externalLoadBalancerDeviceDao.update(lbDevice.getId(), lbDevice);  
+ }  
+ txn.commit();  
+ }  
+ Account account = \_accountDao.findByIdIncludingRemoved(guestConfig.getAccountId());  
+ String action = add ? "implemented" : "shut down";  
+ s\_logger.debug("External load balancer has " + action + " the guest network for account " + account.getAccountName() + "(id = " + account.getAccountId() + ") with VLAN tag " + guestVlanTag);  
+ return true;  
+ }  
========================================

bug 11360: make user\_vm\_details generic to all VMs instead of user-only VMs, VM details info will be carried down to hypervisor resource  
  
Project: cloudstack.txt  
Commit: f70958809bc38e49ad92192cfa3f178b6590fe0b  
ClassName: com.cloud.vm.VMInstanceVO.java  
  
 public Map<String, String> getDetails() {  
- return null;  
+ return details;  
+ }  
+ public void setDetail(String name, String value) {  
**+ assert (details != null) : "Did you forget to load the details?";**========================================

fix bug 7722, code complete basic architecture  
  
Project: cloudstack.txt  
Commit: 5c01c42ba7467d6b867babe88884fc8d94f1724b  
ClassName: com.cloud.network.ovs.OvsNetworkManagerImpl.java  
  
+ public long askVlanId(long accountId, long hostId) {  
**+ assert \_isEnabled : "Who call me ??? while OvsNetwokr is not enabled!!!";**+ final Transaction txn = Transaction.currentTxn();  
+ txn.start();  
+ List<VlanMappingVO> mappings = \_vlanMappingDao.listByAccountIdAndHostId(accountId, hostId);  
+ long vlan = 0;  
+ if (mappings.size() !=0) {  
**+ assert mappings.size() == 1 : "We should only have one vlan for an account on a host";**+ txn.commit();  
+ vlan = mappings.get(0).getVlan();  
+ s\_logger.debug("Already has an Vlan " + vlan + " on host " + hostId  
+ + " for account " + accountId + ", use it!");  
+ return vlan;  
+ }  
+ mappings = \_vlanMappingDao.listByHostId(hostId);  
+ if (mappings.size() > 0) {  
+ ArrayList<Long> vlans = new ArrayList<Long>();  
+ for (VlanMappingVO vo : mappings) {  
+ vlans.add(new Long(vo.getVlan()));  
+ }  
+ int i;  
+ for (i=0; i<4096; i++) {  
+ if (!vlans.contains(new Long(i))) {  
+ vlan = i;  
+ break;  
+ }  
+ }  
**+ assert i!=4096 : "Terrible, vlan exhausted on this server!!!";**+ }  
+ VlanMappingVO newVlan = new VlanMappingVO(accountId, hostId, vlan);  
+ \_vlanMappingDao.persist(newVlan);  
+ \_vlanMappingDirtyDao.markDirty(accountId);  
+ txn.commit();  
+ return 0;  
+ }  
========================================

fix bug 7722, code complete basic architecture  
  
Project: cloudstack.txt  
Commit: 5c01c42ba7467d6b867babe88884fc8d94f1724b  
ClassName: com.cloud.network.ovs.OvsNetworkManagerImpl.java  
  
+ public void applyDefaultFlow(Commands cmds,  
+ VirtualMachineProfile<UserVmVO> profile, DeployDestination dest) {  
+ if (!\_isEnabled) {  
+ return;  
+ }  
+ UserVmVO userVm = profile.getVirtualMachine();  
+ VirtualMachine.Type vmType = userVm.getType();  
+ if (vmType != VirtualMachine.Type.User  
+ && vmType != VirtualMachine.Type.DomainRouter) {  
+ return;  
+ }  
+ List<NicVO> nics = \_nicDao.listBy(userVm.getId());  
+ if (nics.size() == 0)  
+ return;  
+ NicVO nic = null;  
+ if (vmType == VirtualMachine.Type.DomainRouter) {  
+ for (NicVO n : nics) {  
+ NetworkVO network = \_networkDao.findById(n.getNetworkId());  
+ if (network.getTrafficType() == TrafficType.Guest) {  
+ nic = n;  
+ break;  
+ }  
+ }  
+ } else {  
+ nic = nics.get(0);  
+ }  
**+ assert nic!=null : "Why there is no guest network nic???";**+ String vlans = parseVlanAndMapping(nic.getBroadcastUri().toASCIIString());  
+ cmds.addCommand(new OvsSetTagAndFlowCommand(userVm.getName(), vlans));  
+ }  
========================================

bug 12787: improve worker VM cleanup in snapshot operation  
  
Project: cloudstack.txt  
Commit: bde008ed7eb82a7dd0b1e11c93fd313294cb6bc6  
ClassName: com.cloud.storage.resource.VmwareSecondaryStorageResourceHandler.java  
  
- public String getWorkerName(VmwareContext context, Command cmd) {  
+ public String getWorkerName(VmwareContext context, Command cmd, int workerSequence) {  
 **assert(cmd.getContextParam("worker") != null);  
+ assert(workerSequence < 2);**+ if(workerSequence == 0)  
 return cmd.getContextParam("worker");  
+ return cmd.getContextParam("worker2");  
 }  
========================================

bug 11114: when release static nat as a part of vm expunge, do searchIncludingRemoved in nics table as the nics are marked as Removed before static nat is released  
status 11114: resolved fixed  
  
Project: cloudstack.txt  
Commit: 05713a20e89b25b3ed84b636b648cfef6e8f26f9  
ClassName: com.cloud.network.NetworkManagerImpl.java  
  
+ public String getIpInNetworkIncludingRemoved(long vmId, long networkId) {  
+ Nic guestNic = getNicInNetworkIncludingRemoved(vmId, networkId);  
**+ assert (guestNic != null && guestNic.getIp4Address() != null) : "Vm doesn't belong to network associated with ipAddress or ip4 address is null";** return guestNic.getIp4Address();  
========================================

bug 10538: avoid duplicated config values insertion when multiple management servers start at the same time  
status 10538: resolved fixed  
  
Project: cloudstack.txt  
Commit: e0c74004f13fb94e9dd11d8e02621838f3f1da88  
ClassName: com.cloud.service.dao.ServiceOfferingDaoImpl.java  
  
+ public ServiceOfferingVO persistDeafultServiceOffering(ServiceOfferingVO offering) {  
**+ assert offering.getUniqueName() != null : "unique name should be set for the service offering";**+ ServiceOfferingVO vo = findByName(offering.getUniqueName());  
+ if (vo != null) {  
+ return vo;  
+ }  
========================================

Fixed error message  
  
Project: cloudstack.txt  
Commit: 47e8d2b29cba829ca41b21cce7a6ea162b970282  
ClassName: com.cloud.network.firewall.FirewallManagerImpl.java  
  
@@ -290,3 +290,3 @@ public void validateFirewallRule(Account caller, IPAddressVO ipAddress, Integer  
 Network network = \_networkMgr.getNetwork(networkId);  
**- assert network != null : "Can't create port forwarding rule as network associated with public ip address is null...how is it possible?";  
+ assert network != null : "Can't create port forwarding rule as network associated with public ip address is null?";**========================================

CS-9919: Support for Nexus Swiches (Cisco Vswitches)  
Description:  
Added handling for error responses from vsm.  
Fixing an issue with port profile creation.  
  
Project: cloudstack.txt  
Commit: 441a811927156017da420564cfc4b041255fb242  
ClassName: com.cloud.utils.cisco.n1kv.vsm.VsmResponse.java  
  
+ private void parseError(Node element) {  
+ Element rpcError = (Element) element;  
+ try {  
**+ assert(rpcError.getNodeName().equalsIgnoreCase("nf:rpc-error"));**+ for (Node node = rpcError.getFirstChild();  
+ node != null; node = node.getNextSibling()) {  
+ if (node.getNodeName().equalsIgnoreCase("nf:error-type")) {  
+ \_type = ErrorType.valueOf(node.getTextContent().trim());  
+ } else if (node.getNodeName().equalsIgnoreCase("nf:error-tag")) {  
+ \_tag = getErrorTag(node.getTextContent().trim());  
+ } else if (node.getNodeName().equalsIgnoreCase("nf:error-severity")) {  
+ \_severity = ErrorSeverity.valueOf(node.getTextContent().trim());  
+ } else if (node.getNodeName().equalsIgnoreCase("nf:error-path")) {  
+ \_path = node.getTextContent();  
+ } else if (node.getNodeName().equalsIgnoreCase("nf:error-message")) {  
+ \_message = node.getTextContent();  
+ } else if (node.getNodeName().equalsIgnoreCase("nf:error-info")) {  
+ \_info = node.getTextContent();  
+ }  
+ }  
+ } catch (DOMException e) {  
+ s\_logger.error("Error parsing the response : " + e.toString());  
+ }  
+ }  
========================================

applied  
http://issues.apache.org/bugzilla/show\_bug.cgi?id=27854  
  
Project: commons-bcel.txt  
Commit: 87ef0ad4e4760c57a6140cfbd42efaf8d628a20b  
ClassName: org.apache.bcel.verifier.structurals.ControlFlowGraph.java  
  
+ public Frame getInFrame() {  
+ Frame org;  
+ InstructionContext jsr = lastExecutionJSR();  
+ org = (Frame) inFrames.get(jsr);  
+ if (org == null){  
**+ throw new AssertionViolatedException("inFrame not set! This:\n"+this+"\nInFrames: '"+inFrames+"'.");**+ }  
========================================

Make methods that create parsers or printers fail early and provide an expressive error messages. Document new behavior in JavaDoc  
  
Project: commons-csv.txt  
Commit: a2fef00ab8950771735bc09793f05fc842a0cd97  
ClassName: org.apache.commons.csv.CSVParser.java  
  
 public static CSVParser parse(String string, final CSVFormat format) throws IOException {  
**+ Assertions.notNull(string, "string");  
+ Assertions.notNull(format, "format");** return new CSVParser(new StringReader(string), format);  
========================================

Make methods that create parsers or printers fail early and provide an expressive error messages. Document new behavior in JavaDoc  
  
Project: commons-csv.txt  
Commit: a2fef00ab8950771735bc09793f05fc842a0cd97  
ClassName: org.apache.commons.csv.CSVParser.java  
  
@@ -180,2 +195,5 @@ public static CSVParser parse(URL url, Charset charset, final CSVFormat format)  
 public CSVParser(final Reader reader, final CSVFormat format) throws IOException {  
**+ Assertions.notNull(reader, "reader");  
+ Assertions.notNull(format, "format");** format.validate();  
========================================

Make methods that create parsers or printers fail early and provide an expressive error messages. Document new behavior in JavaDoc  
  
Project: commons-csv.txt  
Commit: a2fef00ab8950771735bc09793f05fc842a0cd97  
ClassName: org.apache.commons.csv.CSVParser.java  
  
 public static CSVParser parse(File file, final CSVFormat format) throws IOException {  
**+ Assertions.notNull(file, "file");  
+ Assertions.notNull(format, "format");** return new CSVParser(new FileReader(file), format);  
========================================

refactor/cleanup test suite a bit  
apply generic tests to more configurations  
fix a couple of bugs that revealed  
  
Project: commons-dbcp.txt  
Commit: 2b1a8888a2500430da7a6910939dc5dc1ef3bd41  
ClassName: org.apache.commons.dbcp.cpdsadapter.PooledConnectionImpl.java  
  
@@ -148,2 +148,3 @@ public void close()  
 {   
**+ assertOpen();** isClosed = true;  
========================================

Added rollbackAfterValidation property and code to issue a rollback on a  
connection after validation when this property is set to true to eliminate  
Oracle driver exceptions. Default property value is false.  
JIRA: DBCP-116  
Reported and patched by Thomas Fischer  
  
Project: commons-dbcp.txt  
Commit: dd16ef675629bb506dacf06886d14ecbfffdd561  
ClassName: org.apache.commons.dbcp.datasources.InstanceKeyDataSource.java  
  
+ public boolean isRollbackAfterValidation() {  
+ return (this.rollbackAfterValidation);  
+ }  
+ public void setRollbackAfterValidation(boolean rollbackAfterValidation) {  
**+ assertInitializationAllowed();**========================================

refactor/cleanup test suite a bit  
apply generic tests to more configurations  
fix a couple of bugs that revealed  
  
Project: commons-dbcp.txt  
Commit: 2b1a8888a2500430da7a6910939dc5dc1ef3bd41  
ClassName: org.apache.commons.dbcp.cpdsadapter.ConnectionImpl.java  
  
@@ -172,2 +172,3 @@ public void close()  
 {  
**+ assertOpen();** isClosed = true;  
========================================

**fix assertions**  
Project: commons-functor.txt  
Commit: 7edc73183bc407fcb8f4a9a464908fa533020239  
ClassName: org.apache.commons.functor.example.aggregator.list.OwnListImplementationSample.java  
  
@@ -36,7 +38,6 @@ public void sampleCreateOwnList() {  
 CustomListAggregator<Integer> agg = new CustomListAggregator<Integer>(new IntegerSumAggregatorFunction());  
- int eval = agg.evaluate();  
**- assertEquals(eval, 0);  
+ assertNull(agg.evaluate());** agg.add(2);  
 agg.add(3);  
- eval = agg.evaluate();  
+ int eval = agg.evaluate();  
 **assertEquals(eval, 5);**========================================

Added missed synchronization. Submitted by Sandy McArthur. Issue #37226. Thanks to Mayur Naik for discovering and reporting these issues.  
  
Project: commons-pool.txt  
Commit: f6bb9e741861d50947c21d14e035bff544ed5ddb  
ClassName: org.apache.commons.pool.impl.StackObjectPool.java  
  
- public void addObject() throws Exception {  
+ public synchronized void addObject() throws Exception {  
**+ assertOpen();** Object obj = \_factory.makeObject();  
- synchronized(this) {  
 \_numActive++; // A little slimy - must do this because returnObject decrements it.  
========================================

Added missed synchronization to SoftReferenceObjectPool. Submitted by Sandy McArthur. Issue #37228. Thanks to Mayur Naik for discovering and reporting these issues.  
  
Project: commons-pool.txt  
Commit: f3fb84e0389bdff6cd245a893d301b522b4836b0  
ClassName: org.apache.commons.pool.impl.SoftReferenceObjectPool.java  
  
- public int getNumActive() {  
+ public synchronized int getNumActive() {  
**+ assertOpen();** return \_numActive;  
========================================

Added missed synchronization to GenericObjectPool. Submitted by Sandy McArthur. Issue #37227. Thanks to Mayur Naik for discovering and reporting these issues.  
  
Project: commons-pool.txt  
Commit: b599d829468ef47992277f1f97cfcd9f000cf819  
ClassName: org.apache.commons.pool.impl.GenericObjectPool.java  
  
- public Object borrowObject() throws Exception {  
+ public synchronized Object borrowObject() throws Exception {  
**+ assertOpen();** long starttime = System.currentTimeMillis();  
========================================

Added missed synchronization to SoftReferenceObjectPool. Submitted by Sandy McArthur. Issue #37228. Thanks to Mayur Naik for discovering and reporting these issues.  
  
Project: commons-pool.txt  
Commit: f3fb84e0389bdff6cd245a893d301b522b4836b0  
ClassName: org.apache.commons.pool.impl.SoftReferenceObjectPool.java  
  
- public void addObject() throws Exception {  
+ public synchronized void addObject() throws Exception {  
**+ assertOpen();** Object obj = \_factory.makeObject();  
- synchronized(this) {  
 \_numActive++; // A little slimy - must do this because returnObject decrements it.  
- this.returnObject(obj);  
- }  
+ returnObject(obj);  
 }  
========================================

Added missed synchronization to SoftReferenceObjectPool. Submitted by Sandy McArthur. Issue #37228. Thanks to Mayur Naik for discovering and reporting these issues.  
  
Project: commons-pool.txt  
Commit: f3fb84e0389bdff6cd245a893d301b522b4836b0  
ClassName: org.apache.commons.pool.impl.SoftReferenceObjectPool.java  
  
- public int getNumIdle() {  
+ public synchronized int getNumIdle() {  
**+ assertOpen();** return \_pool.size();  
========================================

Edited all JavaDocs in the pool package.  
Fixed spelling mistakes. Improved consistancy.  
Updated examples. Clarified expectations. etc.  
  
Project: commons-pool.txt  
Commit: b8012fca12b20aebc7daacb3a6704df25d349bf8  
ClassName: org.apache.commons.pool.BaseKeyedObjectPool.java  
  
+ protected final boolean isClosed() {  
+ return closed;  
+ }  
**+ protected final void assertOpen() throws IllegalStateException {**+ if(isClosed()) {  
+ throw new IllegalStateException("Pool not open");  
+ }  
========================================

Added missed synchronization to GenericObjectPool. Submitted by Sandy McArthur. Issue #37227. Thanks to Mayur Naik for discovering and reporting these issues.  
  
Project: commons-pool.txt  
Commit: b599d829468ef47992277f1f97cfcd9f000cf819  
ClassName: org.apache.commons.pool.impl.GenericObjectPool.java  
  
- public void addObject() throws Exception {  
+ public synchronized void addObject() throws Exception {  
**+ assertOpen();** Object obj = \_factory.makeObject();  
========================================

**Tidy up and fix assertions**  
Project: creadur-tentacles.txt  
Commit: e96309356c2cb4e5c91dc1c8f04f0aed1d3e7170  
ClassName: org.apache.creadur.tentacles.Files.java  
  
@@ -93,3 +72,4 @@ public static void mkdirs(File file) {  
**- assert file.isDirectory() : "not a directory" + file;**+ final boolean isDirectory = file.isDirectory();  
**+ assert isDirectory : "Not a directory: " + file;** }  
========================================

Added the ability to use the STSClient in an intermediary.  
- There is a new configuration property to disable storing issued tokens per exchange  
- Caching is enabled for OnBehalfOf or ActAs tokens  
Conflicts:  
rt/ws/security/src/main/java/org/apache/cxf/ws/security/policy/interceptors/IssuedTokenInterceptorProvider.java  
  
Project: cxf.txt  
Commit: 01c1f39c02690722fd569efe14c4e0b97fc89aea  
ClassName: org.apache.cxf.ws.security.policy.interceptors.IssuedTokenInterceptorProvider.java  
  
+ private SecurityToken getTokenFromSTS(  
+ Message message,  
+ STSClient client,  
**+ AssertionInfoMap aim,**+ AddressingProperties maps,  
+ IssuedToken itok,  
+ String appliesTo  
+ ) throws Exception {  
+ client.setTrust(getTrust10(aim));  
+ client.setTrust(getTrust13(aim));  
+ client.setTemplate(itok.getRstTemplate());  
+ if (maps == null) {  
+ return client.requestSecurityToken();  
+ } else {  
+ client.setAddressingNamespace(maps.getNamespaceURI());  
+ return client.requestSecurityToken(appliesTo);  
+ }  
+ }  
========================================

Change to throwing exceptions for various policy issues during out processing.  
  
Project: cxf.txt  
Commit: 61eae25d773873d9fc478af6057c0c5fdb26179f  
ClassName: org.apache.cxf.ws.security.wss4j.policyhandlers.AbstractBindingBuilder.java  
  
**+ protected void assertSupportingTokens(Vector<WSEncryptionPart> sigs) {  
+ assertSupportingTokens(findAndAssertPolicy(SP12Constants.SIGNED\_SUPPORTING\_TOKENS));  
+ assertSupportingTokens(findAndAssertPolicy(SP12Constants.ENDORSING\_SUPPORTING\_TOKENS));  
+ assertSupportingTokens(findAndAssertPolicy(SP12Constants.SIGNED\_ENDORSING\_SUPPORTING\_TOKENS));  
+ assertSupportingTokens(findAndAssertPolicy(SP12Constants.SIGNED\_ENCRYPTED\_SUPPORTING\_TOKENS));  
+ assertSupportingTokens(findAndAssertPolicy(SP12Constants.ENDORSING\_ENCRYPTED\_SUPPORTING\_TOKENS));  
+ assertSupportingTokens(findAndAssertPolicy(SP12Constants**+ .SIGNED\_ENDORSING\_ENCRYPTED\_SUPPORTING\_TOKENS));  
**+ assertSupportingTokens(findAndAssertPolicy(SP12Constants.SUPPORTING\_TOKENS));  
+ assertSupportingTokens(findAndAssertPolicy(SP12Constants.ENCRYPTED\_SUPPORTING\_TOKENS));**+ }   
========================================

Change to throwing exceptions for various policy issues during out processing.  
  
Project: cxf.txt  
Commit: 61eae25d773873d9fc478af6057c0c5fdb26179f  
ClassName: org.apache.cxf.ws.security.wss4j.policyhandlers.AbstractBindingBuilder.java  
  
+ public Crypto getCrypto(TokenWrapper wrapper, boolean sign) {  
+ Crypto crypto = (Crypto)message.getContextualProperty(sign   
+ ? SecurityConstants.SIGNATURE\_CRYPTO   
+ : SecurityConstants.ENCRYPT\_CRYPTO);  
+ if (crypto != null) {  
+ return crypto;  
+ }  
+ Object o = message.getContextualProperty(sign   
+ ? SecurityConstants.SIGNATURE\_PROPERTIES   
+ : SecurityConstants.ENCRYPT\_PROPERTIES);  
+ Properties properties = null;  
+ if (o instanceof Properties) {  
+ properties = (Properties)o;  
+ } else if (o instanceof String) {  
+ ResourceManager rm = message.getExchange().get(Bus.class).getExtension(ResourceManager.class);  
+ URL url = rm.resolveResource((String)o, URL.class);  
+ try {  
+ if (url == null) {  
+ url = ClassLoaderUtils.getResource((String)o, this.getClass());  
+ }  
+ if (url != null) {  
+ properties = new Properties();  
+ properties.load(url.openStream());  
+ } else {  
**+ policyNotAsserted(wrapper, "Could not find properties file " + o);**+ }  
+ } catch (IOException e) {  
**+ policyNotAsserted(wrapper, e);**+ }  
+ } else if (o instanceof URL) {  
+ properties = new Properties();  
+ try {  
+ properties.load(((URL)o).openStream());  
+ } catch (IOException e) {  
**+ policyNotAsserted(wrapper, e);**+ }   
+ }  
+ if (properties != null) {  
+ return CryptoFactory.getInstance(properties);  
+ }  
+ return null;  
+ }  
========================================

Fixed AlgorithmSuite bug + enabled a load of streaming ws-security tests following recent fixes in WSS4J  
  
Project: cxf.txt  
Commit: e8d94a36e0f91ffb18da9a5c3bfb22580ab89afb  
ClassName: org.apache.cxf.ws.security.wss4j.PolicyStaxActionInInterceptor.java  
  
**+ private void assertAllAlgorithmSuites(String spNamespace, AssertionInfoMap aim) {  
+ Collection<AssertionInfo> sp11Ais =**+ aim.get(new QName(spNamespace, SPConstants.ALGORITHM\_SUITE));  
+ if (sp11Ais != null) {  
**+ for (AssertionInfo ai : sp11Ais) {  
+ ai.setAsserted(true);  
+ AlgorithmSuite algorithmSuite = (AlgorithmSuite)ai.getAssertion();**+ AlgorithmSuiteType algorithmSuiteType = algorithmSuite.getAlgorithmSuiteType();  
+ String namespace = algorithmSuiteType.getNamespace();  
+ if (namespace == null) {  
+ namespace = spNamespace;  
+ }  
**+ Collection<AssertionInfo> algAis =**+ aim.get(new QName(namespace, algorithmSuiteType.getName()));  
+ if (algAis != null) {  
**+ for (AssertionInfo algAi : algAis) {  
+ algAi.setAsserted(true);**+ }  
+ }  
+ }  
+ }  
========================================

Some minor security fixes  
  
Project: cxf.txt  
Commit: 796d2adc6b872ccc854ae45eaad1b1d2478f7b6f  
ClassName: org.apache.cxf.ws.security.wss4j.policyvalidators.AbstractBindingPolicyValidator.java  
  
**+ protected void notAssertPolicy(AssertionInfoMap aim, Assertion token, String msg) {  
+ Collection<AssertionInfo> ais = aim.get(token.getName());**+ if (ais != null && !ais.isEmpty()) {  
**+ for (AssertionInfo ai : ais) {  
+ if (ai.getAssertion() == token) {  
+ ai.setNotAsserted(msg);**+ }  
+ }   
+ }  
========================================

[CXF-3649] If its an application level fault, make sure policies are  
considered.  
  
Project: cxf.txt  
Commit: aaae6c5280db12189ee89ae00b0e9def8ce52232  
ClassName: org.apache.cxf.ws.policy.PolicyEngineImpl.java  
  
@@ -241,3 +251,3 @@ public EffectivePolicy getEffectiveServerFaultPolicy(EndpointInfo ei, BindingFau  
 }  
**- epi.initialise(ei, bfi, this, assertor);  
+ epi.initialise(ei, boi, bfi, this, assertor);** bfi.setProperty(POLICY\_INFO\_FAULT\_SERVER, epi);  
========================================

Squashed commit of the following:  
commit 4699706a51f6e75641be257089636aa0dbe7980d  
Author: Daniel Kulp <dkulp@apache.org>  
Date: Mon Apr 4 14:16:07 2011 -0400  
Use a wsdl for the sample sts so ?wsdl provides a useful contract.  
Also will provice a place to put the WS-SecPol requirements  
commit 3a44f27efb80bb0211115df7284b84860a1e3de2  
Author: Zsolt Beothy-Elo <zbeothy-elo@talend.com>  
Date: Sat Apr 2 11:13:17 2011 +0200  
Include sts sample in build.  
commit e522cd0f5a5d8c99aa8b903d24a38e7b13ad65ea  
Author: Zsolt Beothy-Elo <zbeothy-elo@talend.com>  
Date: Sat Apr 2 09:34:07 2011 +0200  
Move sts sample sources to demo.sts...  
commit b33ad2b58458a4154fdffad808b32aa089081896  
Author: Zsolt Beothy-Elo <zbeothy-elo@talend.com>  
Date: Sat Apr 2 08:27:04 2011 +0200  
Add spring configuration and enhance pom for sts sample  
commit 1ffc68a7cc3bd4d092e55cd7e4a8896c618510e5  
Author: Zsolt Beothy-Elo <zbeothy-elo@talend.com>  
Date: Fri Apr 1 19:27:18 2011 +0200  
Move remaining sample specific classes from the cxf security module to the sts\_issue\_operation sample module  
commit 4a5df71944b8e2fc76a8bf45cb2cf19efe6dc8af  
Author: Vladimir Romaniuk <vromaniuk@talend.com>  
Date: Fri Apr 1 12:53:56 2011 +0300  
added keystore and test modified according changes in keystore  
commit f3265f4df54e1acd6ce5bd9f894015b8ee35b376  
Author: Vladimir Romaniuk <vromaniuk@talend.com>  
Date: Fri Apr 1 12:22:55 2011 +0300  
Sample issue operation implementation  
commit 5526ec67bfbd7946d6b8d0732c60c5864f36d1f8  
Author: Daniel Kulp <dkulp@apache.org>  
Date: Fri Mar 25 14:29:30 2011 -0400  
First pass at STS from Anubhav Sharma  
  
Project: cxf.txt  
Commit: e980b4340dc82835b056f24e6844e3adfea8bd85  
ClassName: demo.sts.provider.operation.impl.SAMLTokenIssueOperation.java  
  
**+ private RequestSecurityTokenResponseType wrapAssertionToResponse(  
+ String tokenType, Element samlAssertion, String tokenId) {**+ RequestSecurityTokenResponseType response = WS\_TRUST\_FACTORY  
+ .createRequestSecurityTokenResponseType();  
+ JAXBElement<String> jaxbTokenType = WS\_TRUST\_FACTORY  
+ .createTokenType(tokenType);  
+ response.getAny().add(jaxbTokenType);  
+ RequestedSecurityTokenType requestedTokenType = WS\_TRUST\_FACTORY  
+ .createRequestedSecurityTokenType();  
+ JAXBElement<RequestedSecurityTokenType> requestedToken = WS\_TRUST\_FACTORY  
+ .createRequestedSecurityToken(requestedTokenType);  
**+ requestedTokenType.setAny(samlAssertion);**+ response.getAny().add(requestedToken);  
+ RequestedReferenceType requestedReferenceType = WS\_TRUST\_FACTORY  
+ .createRequestedReferenceType();  
+ SecurityTokenReferenceType securityTokenReferenceType = WSSE\_FACTORY  
+ .createSecurityTokenReferenceType();  
+ KeyIdentifierType keyIdentifierType = WSSE\_FACTORY  
+ .createKeyIdentifierType();  
+ keyIdentifierType.setValue(tokenId);  
+ JAXBElement<KeyIdentifierType> keyIdentifier = WSSE\_FACTORY  
+ .createKeyIdentifier(keyIdentifierType);  
+ securityTokenReferenceType.getAny().add(keyIdentifier);  
+ requestedReferenceType  
+ .setSecurityTokenReference(securityTokenReferenceType);  
+ JAXBElement<RequestedReferenceType> requestedAttachedReference = WS\_TRUST\_FACTORY  
+ .createRequestedAttachedReference(requestedReferenceType);  
+ response.getAny().add(requestedAttachedReference);  
+ JAXBElement<RequestedReferenceType> requestedUnattachedReference = WS\_TRUST\_FACTORY  
+ .createRequestedUnattachedReference(requestedReferenceType);  
+ response.getAny().add(requestedUnattachedReference);  
+ return response;  
+ }  
========================================

Another 300+ backportable rawtypes warning fixes  
  
Project: cxf.txt  
Commit: f196317cf31e3c0cfdd050d1f269203fe6cb0ea4  
ClassName: org.apache.cxf.ws.security.policy.builders.AsymmetricBindingBuilder.java  
  
**@@ -68,4 +68,4 @@ public Assertion build(Element element, AssertionBuilderFactory factory)**- for (Iterator iterator = policy.getAlternatives(); iterator.hasNext();) {  
- processAlternative((List)iterator.next(), asymmetricBinding, consts);  
**+ for (Iterator<List<Assertion>> iterator = policy.getAlternatives(); iterator.hasNext();) {**+ processAlternative(iterator.next(), asymmetricBinding, consts);  
========================================

Fixed AlgorithmSuite bug + enabled a load of streaming ws-security tests following recent fixes in WSS4J  
  
Project: cxf.txt  
Commit: e8d94a36e0f91ffb18da9a5c3bfb22580ab89afb  
ClassName: org.apache.cxf.ws.security.policy.custom.DefaultAlgorithmSuiteLoader.java  
  
**+ protected void parseCustomAssertion(Assertion assertion) {  
+ String assertionName = assertion.getName().getLocalPart();  
+ String assertionNamespace = assertion.getName().getNamespaceURI();  
+ if (!"http://cxf.apache.org/custom/security-policy".equals(assertionNamespace)) {**+ return;  
+ }  
**+ if ("Basic128GCM".equals(assertionName)) {**+ setAlgorithmSuiteType(algorithmSuiteTypes.get("Basic128GCM"));  
**+ getAlgorithmSuiteType().setNamespace(assertionNamespace);  
+ } else if ("Basic192GCM".equals(assertionName)) {**+ setAlgorithmSuiteType(algorithmSuiteTypes.get("Basic192GCM"));  
**+ getAlgorithmSuiteType().setNamespace(assertionNamespace);  
+ } else if ("Basic256GCM".equals(assertionName)) {**+ setAlgorithmSuiteType(algorithmSuiteTypes.get("Basic256GCM"));  
 **getAlgorithmSuiteType().setNamespace(assertionNamespace);**========================================

Fix the split package issue for org.apache.cxf.ws.addressing  
Move impls into impl subpackage, put some stubs in addressing  
Make sure more classes don't use the impls (still some in RM)  
  
Project: cxf.txt  
Commit: f19c9841f5cd356cbfc3ed3c60f6fcbaf5c97290  
ClassName: org.apache.cxf.ws.addressing.impl.MAPAggregatorImpl.java  
  
**+ private void assertAddressing(Message message) {  
+ AssertionInfoMap aim = message.get(AssertionInfoMap.class);**+ if (null == aim) {  
+ return;  
+ }  
+ QName[] types = new QName[] {  
**+ MetadataConstants.ADDRESSING\_ASSERTION\_QNAME, MetadataConstants.USING\_ADDRESSING\_2004\_QNAME,**+ MetadataConstants.USING\_ADDRESSING\_2005\_QNAME, MetadataConstants.USING\_ADDRESSING\_2006\_QNAME  
+ };  
+ for (QName type : types) {  
**+ assertAssertion(aim, type);  
+ if (type.equals(MetadataConstants.ADDRESSING\_ASSERTION\_QNAME)) {  
+ assertAssertion(aim, MetadataConstants.ANON\_RESPONSES\_ASSERTION\_QNAME);  
+ assertAssertion(aim, MetadataConstants.NON\_ANON\_RESPONSES\_ASSERTION\_QNAME);  
+ } else if (type.equals(MetadataConstants.ADDRESSING\_ASSERTION\_QNAME\_0705)) {  
+ assertAssertion(aim, MetadataConstants.ANON\_RESPONSES\_ASSERTION\_QNAME\_0705);  
+ assertAssertion(aim, MetadataConstants.NON\_ANON\_RESPONSES\_ASSERTION\_QNAME\_0705);**+ }  
+ }  
+ }  
========================================

[CXF-4264] Applied fix from Juerg. Thanks. Some minor fixes and refactoring  
  
Project: cxf-fediz.txt  
Commit: 287615305e037fe03a3c7de924777a4ca9f421b1  
ClassName: org.apache.cxf.fediz.core.saml.SAMLTokenValidator.java  
  
**- protected List<Claim> parseClaimsInAssertion(org.opensaml.saml2.core.Assertion assertion) {**- List<org.opensaml.saml2.core.AttributeStatement> attributeStatements =   
**- assertion.getAttributeStatements();  
+ protected List<Claim> parseClaimsInAssertion(  
+ org.opensaml.saml2.core.Assertion assertion) {  
+ List<org.opensaml.saml2.core.AttributeStatement> attributeStatements = assertion**+ .getAttributeStatements();  
 if (attributeStatements == null || attributeStatements.isEmpty()) {  
========================================

[CXF-4264] Applied fix from Juerg. Thanks. Some minor fixes and refactoring  
  
Project: cxf-fediz.txt  
Commit: 287615305e037fe03a3c7de924777a4ca9f421b1  
ClassName: org.apache.cxf.fediz.core.saml.SAMLTokenValidator.java  
  
@@ -153,8 +160,4 @@ public TokenValidatorResponse validateAndProcessToken(Element token, FederationC  
 TokenValidatorResponse response = new TokenValidatorResponse(  
**- assertion.getId(),**- p.getName(),  
**- assertionIssuer,**- roles,  
- claims,  
- audience);  
**+ assertion.getId(), p.getName(), assertionIssuer, roles,**+ claims, audience);  
========================================

[CXF-4264] Applied fix from Juerg. Thanks. Some minor fixes and refactoring  
  
Project: cxf-fediz.txt  
Commit: 287615305e037fe03a3c7de924777a4ca9f421b1  
ClassName: org.apache.cxf.fediz.core.saml.SAMLTokenValidator.java  
  
**- protected String getAudienceRestriction(org.opensaml.saml2.core.Assertion assertion) {**+ protected String getAudienceRestriction(  
**+ org.opensaml.saml2.core.Assertion assertion) {** String audience = null;  
 try {  
**- audience = assertion.getConditions().getAudienceRestrictions().get(0).getAudiences().get(0).getAudienceURI();  
+ audience = assertion.getConditions().getAudienceRestrictions()**+ .get(0).getAudiences().get(0).getAudienceURI();  
 } catch (Exception ex) {  
========================================

Finish ReviewMgr REST support. Fix bugs kicked up from the regression tests.  
  
Project: directory-fortress-core.txt  
Commit: aa36af60ad8e87c6c34dab29178fec24f9bc73b3  
ClassName: com.jts.fortress.rest.ReviewMgrRestImpl.java  
  
@@ -495,4 +601,22 @@ public final User readUser(User user)  
 {  
- throw new java.lang.UnsupportedOperationException();  
**+ VUtil.assertNotNullOrEmpty(userId, GlobalErrIds.USER\_NULL, OCLS\_NM + ".assignedRoles");**+ List<String> retUserRoles;  
+ FortRequest request = new FortRequest();  
+ request.setValue(userId);  
+ if (this.adminSess != null)  
+ {  
+ request.setSession(adminSess);  
+ }  
+ String szRequest = RestUtils.marshal(request);  
+ String szResponse = RestUtils.post(USERID, PW, szRequest, Ids.Services.roleAsigned.toString());  
+ FortResponse response = RestUtils.unmarshall(szResponse);  
+ if (response.getErrorCode() == 0)  
+ {  
+ retUserRoles = response.getValues();  
+ }  
+ else  
+ {  
+ throw new SecurityException(response.getErrorCode(), response.getErrorMessage());  
+ }  
+ return retUserRoles;  
 }  
========================================

Fixed DIRSERVER-823 (proting the fix from 1.0.1)  
Added a test for it  
  
Project: directory-shared.txt  
Commit: 565704ecf9a988953f1b8a9d68f6cdab91d75f66  
ClassName: org.apache.directory.shared.ldap.codec.actions.InitApproxMatchFilterAction.java  
  
@@ -61,3 +61,5 @@ public void action( IAsn1Container container ) throws DecoderException  
**- Filter filter = new AttributeValueAssertionFilter( LdapConstants.APPROX\_MATCH\_FILTER );  
+ Filter filter = new AttributeValueAssertionFilter(**+ ldapMessageContainer.getTlvId(),  
+ LdapConstants.APPROX\_MATCH\_FILTER );  
========================================

Fixed DIRSERVER-823 (proting the fix from 1.0.1)  
Added a test for it  
  
Project: directory-shared.txt  
Commit: 565704ecf9a988953f1b8a9d68f6cdab91d75f66  
ClassName: org.apache.directory.shared.ldap.codec.actions.InitLessOrEqualFilterAction.java  
  
@@ -64,3 +64,5 @@ public void action( IAsn1Container container ) throws DecoderException  
**- Filter filter = new AttributeValueAssertionFilter( LdapConstants.LESS\_OR\_EQUAL\_FILTER );  
+ Filter filter = new AttributeValueAssertionFilter(**+ ldapMessageContainer.getTlvId(),  
+ LdapConstants.LESS\_OR\_EQUAL\_FILTER );  
========================================

got rid of compiler issue in IDEA  
  
Project: directory-shared.txt  
Commit: ebfeea6b6fd01e67f83bc5876e7e9c54d689241e  
ClassName: org.apache.directory.shared.ldap.message.LdapEncoder.java  
  
@@ -618,3 +618,5 @@ private int computeCompareRequestLength( CompareRequestImpl compareRequest )  
**- if ( compareRequest.getAssertionValue() instanceof BinaryValue)  
+ org.apache.directory.shared.ldap.model.entry.Value assertionValue = compareRequest.getAssertionValue();  
+ if ( assertionValue instanceof BinaryValue )** {  
========================================

Forgot to commit new files from Stefan's latest patch, my fault.  
  
Project: directory-shared.txt  
Commit: 39d50e89fe8c3b35937eaa11c52938c3b0e07ef2  
ClassName: org.apache.directory.shared.ldap.schema.syntax.DITStructureRuleDescriptionSyntaxChecker.java  
  
**+ public void assertSyntax( Object value ) throws NamingException**+ {  
+ if ( !isValidSyntax( value ) )  
+ {  
+ throw new LdapInvalidAttributeValueException( ResultCodeEnum.INVALID\_ATTRIBUTE\_SYNTAX );  
+ }  
+ }  
========================================

Fixed DIRSERVER-823 (proting the fix from 1.0.1)  
Added a test for it  
  
Project: directory-shared.txt  
Commit: 565704ecf9a988953f1b8a9d68f6cdab91d75f66  
ClassName: org.apache.directory.shared.ldap.codec.actions.InitGreaterOrEqualFilterAction.java  
  
@@ -64,3 +64,5 @@ public void action( IAsn1Container container ) throws DecoderException  
**- Filter filter = new AttributeValueAssertionFilter( LdapConstants.GREATER\_OR\_EQUAL\_FILTER );  
+ Filter filter = new AttributeValueAssertionFilter(**+ ldapMessageContainer.getTlvId(),  
+ LdapConstants.GREATER\_OR\_EQUAL\_FILTER );  
========================================

Fix checkstyle errors: HiddenField and Javadoc  
  
Project: directory-shared.txt  
Commit: 78e625c904c71603d6c8a2b0539ba6d06f83f163  
ClassName: org.apache.directory.shared.ldap.message.CompareRequestImpl.java  
  
**@@ -144,9 +142,9 @@ public void setAssertionValue( String attrVal )  
- public void setAssertionValue( byte[] attrVal )  
+ public void setAssertionValue( byte[] value )** {  
- if ( attrVal != null )  
+ if ( value != null )  
 {  
- this.attrVal = new BinaryValue( attrVal );  
+ this.attrVal = new BinaryValue( value );  
 }  
========================================

Forgot to commit new files from Stefan's latest patch, my fault.  
  
Project: directory-shared.txt  
Commit: 39d50e89fe8c3b35937eaa11c52938c3b0e07ef2  
ClassName: org.apache.directory.shared.ldap.schema.syntax.NameFormDescriptionSyntaxChecker.java  
  
**+ public void assertSyntax( Object value ) throws NamingException**+ {  
+ if ( !isValidSyntax( value ) )  
+ {  
+ throw new LdapInvalidAttributeValueException( ResultCodeEnum.INVALID\_ATTRIBUTE\_SYNTAX );  
+ }  
+ }  
========================================

Ok fixed the message hierachy. Problem is we were trying to reuse too much.  
The message hiearchy in the dsml parser uses the message hierarchy in the  
codec, which uses the message hierarchy in the model. The stuff in the dsml  
parser was not using the right hierarchy so I fixed that.  
  
Project: directory-shared.txt  
Commit: aa88a9e24794220c1da2f572b55c47431ec2f7ac  
ClassName: org.apache.directory.shared.dsmlv2.request.CompareRequestDsml.java  
  
**@@ -142,3 +143,3 @@ public void setAssertionValue( Object assertionValue )** {  
**- ( ( CompareRequest ) getDecoratedMessage() ).setAssertionValue( ( byte[] ) assertionValue );  
+ ( ( CompareRequest ) getDecorated() ).setAssertionValue( ( byte[] ) assertionValue );** }  
========================================

Fixed DIRSERVER-823 (proting the fix from 1.0.1)  
Added a test for it  
  
Project: directory-shared.txt  
Commit: 565704ecf9a988953f1b8a9d68f6cdab91d75f66  
ClassName: org.apache.directory.shared.ldap.codec.actions.InitEqualityMatchFilterAction.java  
  
@@ -64,3 +64,5 @@ public void action( IAsn1Container container ) throws DecoderException  
**- Filter filter = new AttributeValueAssertionFilter( LdapConstants.EQUALITY\_MATCH\_FILTER );  
+ Filter filter = new AttributeValueAssertionFilter(**+ ldapMessageContainer.getTlvId(),  
+ LdapConstants.EQUALITY\_MATCH\_FILTER );  
========================================

Fix checkstyle errors: naming conventions and Javadocs  
  
Project: directory-shared.txt  
Commit: e9cd76d6c15cd3e091f5cbd189f652799c11e593  
ClassName: org.apache.directory.shared.ldap.codec.LdapMessageGrammar.java  
  
@@ -2668,5 +2668,5 @@ public void action( Asn1Container container )  
- if ( log.isDebugEnabled() )  
+ if ( LOG.isDebugEnabled() )  
 {  
**- log.debug( "Comparing attribute value {}", compareRequest.getAssertionValue() );  
+ LOG.debug( "Comparing attribute value {}", compareRequest.getAssertionValue() );** }  
========================================

Resolving issue DIRSTUDIO-14. The parser now works perfectly if a value is empty (<value></value>). A source formating has also been done to match code template requirements.  
  
Project: directory-studio.txt  
Commit: 595b3b10d2d75e27ac4e348c39f558c0907a8abd  
ClassName: org.apache.directory.ldapstudio.dsmlv2.Dsmlv2Grammar.java  
  
@@ -2321,3 +2313,4 @@ public void action( Dsmlv2Container container ) throws XmlPullParserException  
**- AttributeValueAssertionFilter filter = new AttributeValueAssertionFilter( LdapConstants.LESS\_OR\_EQUAL\_FILTER );  
+ AttributeValueAssertionFilter filter = new AttributeValueAssertionFilter(**+ LdapConstants.LESS\_OR\_EQUAL\_FILTER );  
========================================

DRILL-620: Memory consumption fixes  
accounting fixes  
trim buffers  
switch to using setSafe and copySafe methods only  
adaptive allocation  
operator based allocator wip  
handle OOM  
Operator Context  
  
Project: drill.txt  
Commit: a2355d42dbff51b858fc28540915cf793f1c0fac  
ClassName: org.apache.drill.exec.physical.impl.join.HashJoinProbeTemplate.java  
  
@@ -96,3 +96,4 @@ public void executeProjectRightPhase() {  
 while (outputRecords < RecordBatch.MAX\_BATCH\_SIZE && recordsProcessed < recordsToProcess) {  
- projectBuildRecord(unmatchedBuildIndexes.get(recordsProcessed++), outputRecords++);  
+ boolean success = projectBuildRecord(unmatchedBuildIndexes.get(recordsProcessed++), outputRecords++);  
**+ assert success;** }  
========================================

Update typing system. Update RPC system. Add Fragmenting Implementation. Working single node. Distributed failing due to threading issues.  
  
Project: drill.txt  
Commit: e57a8d6d4ae282a79bc6e0a7354de992c391300f  
ClassName: org.apache.drill.exec.physical.impl.SimpleFragmentRun.java  
  
+ public void runNoExchangeFragment() throws Exception {  
+ try(RemoteServiceSet serviceSet = RemoteServiceSet.getLocalServiceSet();   
+ Drillbit bit = new Drillbit(CONFIG, serviceSet);   
+ DrillClient client = new DrillClient(CONFIG, serviceSet.getCoordinator());){  
+ bit.run();  
+ client.connect();  
+ List<QueryResultBatch> results = client.runQuery(QueryType.PHYSICAL, Files.toString(FileUtils.getResourceAsFile("/physical\_test2.json"), Charsets.UTF\_8));  
+ RecordBatchLoader batchLoader = new RecordBatchLoader(bit.getContext().getAllocator());  
+ int recordCount = 0;  
+ for (QueryResultBatch batch : results) {  
+ if(!batch.hasData()) continue;  
+ boolean schemaChanged = batchLoader.load(batch.getHeader().getDef(), batch.getData());  
+ boolean firstColumn = true;  
+ if (schemaChanged) {  
+ System.out.println("\n\n========NEW SCHEMA=========\n\n");  
+ for (IntObjectCursor<ValueVector<?>> v : batchLoader) {  
+ if (firstColumn) {  
+ firstColumn = false;  
+ } else {  
+ System.out.print("\t");  
+ }  
+ System.out.print(v.value.getField().getName());  
+ System.out.print("[");  
+ System.out.print(v.value.getField().getType().getMinorType());  
+ System.out.print("]");  
+ }  
+ System.out.println();  
+ }  
+ for (int i = 0; i < batchLoader.getRecordCount(); i++) {  
+ boolean first = true;  
+ recordCount++;  
+ for (IntObjectCursor<ValueVector<?>> v : batchLoader) {  
+ if (first) {  
+ first = false;  
+ } else {  
+ System.out.print("\t");  
+ }  
+ System.out.print(v.value.getObject(i));  
+ }  
+ if(!first) System.out.println();  
+ }  
+ }  
+ logger.debug("Received results {}", results);  
**+ assertEquals(recordCount, 200);**+ }  
+ }  
========================================

DRILL-1309: Implement ProjectPastFilterPushdown and update DrillScanRel cost model so that exclusive column so that star query is more expensive than exclusive column projection. Various fixes affecting record reaaders to handle `\*` column as well as fixes to some test cases.  
exclude parquet files from rat check  
  
Project: drill.txt  
Commit: f148694738a84832d75aca4ef69bff47c68b463f  
ClassName: org.apache.drill.exec.store.AbstractRecordReader.java  
  
+ protected final void setColumns(Collection<SchemaPath> projected) {  
**+ assert Preconditions.checkNotNull(projected, COL\_NULL\_ERROR).size() > 0 : COL\_EMPTY\_ERROR;**+ isStarQuery = isStarQuery(projected);  
+ columns = transformColumns(projected);  
+ }  
========================================

DRILL-1202: fixes memory leak issues: i) ProducerConsumerBatch should clean up resources ii) FragmentExecutor should clean up gracefully at faulty & non-faulty runs regardless  
  
Project: drill.txt  
Commit: 28fbdad886583bc7193833abb06deea327ec8fbe  
ClassName: org.apache.drill.exec.work.fragment.FragmentExecutor.java  
  
@@ -121,21 +110,10 @@ public void run() {  
 }  
- }  
 }  
- root.stop();  
- if(context.isFailed()) {  
- internalFail(context.getFailureCause());  
 }  
- closed = true;  
- context.close();  
**- }catch(AssertionError | Exception ex){**- logger.debug("Caught exception while running fragment", ex);  
- internalFail(ex);  
**+ } catch (AssertionError | Exception e) {**+ logger.debug("Failure while initializing operator tree", e);  
+ context.fail(e);  
+ internalFail(e);  
 } finally {  
- Thread.currentThread().setName(originalThread);  
- if(!closed) {  
- try {  
========================================

DRILL-945: Implementation of repeated reader and writer for parquet. Includes a fairly substantial refactoring of the overall reader structure.  
Fix records counts expected in parquet read tests, previously the wrong records counts were being expected and messages were sent to the logger, but in the form of debug messages. not errors, so they were not flagging when the tests were failing.  
After review: Rename PageReadStatus to PageReader, removed underscore from package name column\_readers, address review comment from parth in VarLengthColumnReaders.  
Fix regression in nullable varlength reader for parquet. Change output names in parquet writer/reader tests to prevent issues with files not being deleted correctly during tests.  
  
Project: drill.txt  
Commit: 5b73c2140f100bb74cbf4a20cd31d13ebf5a4b88  
ClassName: org.apache.drill.exec.store.parquet.columnreaders.VarLengthValuesColumn.java  
  
+ protected void readField(long recordToRead) {  
+ dataTypeLengthInBits = variableWidthVector.getAccessor().getValueLength(valuesReadInCurrentPass);  
+ boolean success = setSafe((int) valuesReadInCurrentPass, pageReader.pageDataByteArray,  
+ (int) pageReader.readPosInBytes + 4, dataTypeLengthInBits);  
**+ assert success;**+ updatePosition();  
+ }  
========================================

**Improve error messaging for OperatorStats assertion messages.**  
Project: drill.txt  
Commit: d7a75ddaf9e2f2e998b071fc1f3fad1b3579f01c  
ClassName: org.apache.drill.exec.ops.OperatorStats.java  
  
 public void startWait() {  
**- assert !inWait;  
+ assert !inWait : "Failure while starting waiting. Currently in waiting.";** stopProcessing();  
========================================

**Improve error messaging for OperatorStats assertion messages.**  
Project: drill.txt  
Commit: d7a75ddaf9e2f2e998b071fc1f3fad1b3579f01c  
ClassName: org.apache.drill.exec.ops.OperatorStats.java  
  
 public void startProcessing() {  
**- assert !inProcessing;  
+ assert !inProcessing : "Failure while starting processing. Currently in processing.";** processingMark = System.nanoTime();  
========================================

Update typing system. Update RPC system. Add Fragmenting Implementation. Working single node. Distributed failing due to threading issues.  
  
Project: drill.txt  
Commit: e57a8d6d4ae282a79bc6e0a7354de992c391300f  
ClassName: org.apache.drill.exec.rpc.bit.BitTunnel.java  
  
+ public final void run() {  
+ try {  
+ BitConnection connection = manager.getConnection(0);  
**+ assert connection != null : "The connection manager should never return a null connection. Worse case, it should throw an exception.";**+ CheckedFuture<T, RpcException> rpc = doRpcCall(connection);  
+ rpc.addListener(new FutureBridge<T>((SettableFuture<T>) delegate(), rpc), MoreExecutors.sameThreadExecutor());  
+ } catch (RpcException ex) {  
+ ((SettableFuture<T>) delegate()).setException(ex);  
+ }  
+ }  
========================================

DRILL-1329: External sort memory fixes  
  
Project: drill.txt  
Commit: 437366f03e5de4c4692703007bff2f5a134720dd  
ClassName: org.apache.drill.exec.physical.impl.xsort.PriorityQueueCopierTemplate.java  
  
@@ -83,4 +79,3 @@ public int next() {  
 int batch = compoundIndex >>> 16;  
**- assert batch < batchGroups.size() \* 2 : String.format("batch: %d batchGroups: %d", batch, batchGroups.size());**- int batchGroup = batch / 2;  
**+ assert batch < batchGroups.size() : String.format("batch: %d batchGroups: %d", batch, batchGroups.size());** if (!doCopy(compoundIndex, outgoingIndex)) {  
========================================

DRILL-1385, along with some cleanup  
Cleaned up option handling. This includes using finals, making member variables  
private whenever possible, and some formatting.  
- fixed a bug in the string formatting for the double range validator  
- OptionValidator, OptionValue, and their implementations now conspire not to  
allow the creation of malformed options because the OptionType has been added  
to validator calls to handle OptionValues that are created on demand.  
Started with updated byte code rewrite from Jacques  
Fixed several problems with scalar value replacement:  
- use consistent ASM api version throughout  
- stop using deprecated ASM methods (actually causes bugs)  
- visitMethodInsn()  
- added a couple of missing super.visitEnd()s  
- fixed a couple of minor FindBugs issues  
- accounted for required stack size increases when replacing holders for  
longs and doubles  
- added accounting for frame offsets to cope with long and double local  
variables and value holder members  
- fixed a few minor bugs found with FindBugs  
- stop using carrotlabs' hash map lget() method on shared constant data  
- fixed an incorrect use of DUP2 on objectrefs when copying long or double  
holder members into locals  
- fixed a problem with redundant POP instructions left behind after replacement  
- fixed a problem with incorrect DUPs in multiple assignment statements  
- fixed a problem with DUP\_X1 replacement when handling constants in multiple  
assignment statements  
- fixed a problem with non-replaced holder member post-decrements  
- don't replace holders passed to static functions as "out" parameters  
(common with Accessors on repeated value vectors)  
- increased the maximum required stack size when transferring holder members to  
locals  
- changed the code generation block type mappings for constants for external  
sorts  
- fixed problems handling constant and non-constant member variables in  
operator classes  
- in general, if a holder is assigned to or from an operator member variable,  
it can't be replaced (at least not until we replace those as well)  
- Use a derived ASM Analyzer (MethodAnalyzer) and Frame  
(AssignmentTrackingFrame) in order to establish relationships between  
assignments of holders through chains of local variables. This effectively  
back-propagates non-replaceability attributes so that if a holder variable  
that can't be replaced is assigned to from another holder variable, that  
second one cannot be replaced either, and so on through longer chains of  
assignments.  
- code for dumping generated source code  
- MergeAdapter dumps before and after results of scalar replacement  
(if it's on)  
- fixed some problems in ReplacingBasicValue by replacing HashSet with  
IdentityHashMap  
- made loggers private  
- added a retry strategy for scalar replacement  
if a scalar replacement code rewriting fails, then this will try to  
regenerate the bytecode again without the scalar replacement.  
- bytecode verification is always on now (required for the retry strategy)  
- use system option to determine whether scalar replacement should be used  
- default option: if scalar replacement fails, retry without it  
- force replacement on or off  
- unit tests for the retry strategy are based on a single known failure case,  
covered by DRILL-2326.  
- add tests TestConvertFunctions to test the three scalar replacement options  
for the failing test case (testVarCharReturnTripConvertLogical)  
- made it possible to set a SYSTEM option as a java property in Drillbit  
- added a command line argument to force scalar replacement to be on during  
testing in the rootmost pom.xml  
In the course of this, added increased checking of intermediate stages of code  
rewriting, as well as logging of classes that cause failures.  
- work around a bug in ASM's CheckClassAdapter that doesn't allow for checking  
of inner classes  
Added comments, tidied up formatting, and added "final" in a number of places.  
Signed-off-by: vkorukanti <venki.korukanti@gmail.com>  
  
Project: drill.txt  
Commit: 7ea212a26dca9b5eca2719300df27751e73dbbcf  
ClassName: org.apache.drill.exec.compile.bytecode.ReplacingBasicValue.java  
  
+ public void associate(final ReplacingBasicValue other) {  
+ associate0(other);  
**+ if (AssertionUtil.ASSERT\_ENABLED) {  
+ assert associates == other.associates;  
+ assert flagSet == other.flagSet;  
+ assert associates.containsKey(this);  
+ assert associates.containsKey(other);**+ for(ReplacingBasicValue value : associates.keySet()) {  
**+ assert associates.get(value) == null; // we never use the value  
+ assert value.associates == associates;  
+ assert value.flagSet == flagSet;**+ }  
+ }  
+ }  
========================================

FALCON-563 Fix checkstyle issues of falcon-regression contributed by Raghav Kumar Gautam  
  
Project: falcon.txt  
Commit: ad51ca03f3af9fe7dcec47291c9f7a6ea586cb1d  
ClassName: org.apache.falcon.regression.core.interfaces.IEntityManagerHelper.java  
  
@@ -110,4 +111,4 @@ public HCatClient getHCatClient() {  
 } catch (HCatException e) {  
**- Assert.fail("Unable to create hCatClient because of exception:\n" +**- ExceptionUtils.getStackTrace(e));  
**+ Assert.fail("Unable to create hCatClient because of exception:\n"**+ + ExceptionUtils.getStackTrace(e));  
 }  
========================================

FALCON-563 Fix checkstyle issues of falcon-regression contributed by Raghav Kumar Gautam  
  
Project: falcon.txt  
Commit: ad51ca03f3af9fe7dcec47291c9f7a6ea586cb1d  
**ClassName: org.apache.falcon.regression.core.util.AssertUtil.java  
 public static void assertFailed(ServiceResponse response) throws JAXBException {  
- Assert.assertNotEquals(response.message, "null", "response message should not be null");  
+ Assert.assertNotEquals(response.getMessage(), "null", "response message should not be null");**========================================

Fixed bugs  
1. Journal error when there are no actions - DE495  
2. Duplicate properties - DE493  
3. Updating only endTime - DE494  
4. return mesaage contains the whole entity - DE487  
5. Upgraded oozie version  
  
Project: falcon.txt  
Commit: cd9adf1967e2cf7bb7511bde1dc919d88e8c0556  
ClassName: org.apache.ivory.converter.OozieProcessMapper.java  
  
@@ -226,6 +231,6 @@ public COORDINATORAPP createLateCoordinator(Cluster cluster, Path bundlePath) th  
 longestOffset = durationLocal;  
- offset = offsetLocal;  
+ offsetExpr = offsetLocal;  
 }  
 }  
**- assert !offset.isEmpty() : "dont expect offset to be empty";  
+ assert !offsetExpr.isEmpty() : "dont expect offset to be empty";**========================================

FALCON-698 Fix checkstyle bugs in test files in falcon-regression. Contributed by Ruslan Ostafiychuk and Raghav Kumar Gautam  
  
Project: falcon.txt  
Commit: 387604d18047acf6923b111bf75f661d94f0f960  
ClassName: org.apache.falcon.regression.core.util.MatrixUtil.java  
  
+ public static Object[][] append(Object[][] arr1, Object[][] arr2) {  
**+ Assert.assertFalse(ArrayUtils.isEmpty(arr1), "arr1 can't be empty:"**+ + Arrays.deepToString(arr1));  
**+ Assert.assertFalse(ArrayUtils.isEmpty(arr2), "arr2 can't be empty:"**+ + Arrays.deepToString(arr2));  
**+ Assert.assertEquals(arr1[0].length, arr2[0].length, "Array rows are not compatible. "**+ + "row of first array: " + Arrays.deepToString(arr1[0])  
+ + "row of second array: " + Arrays.deepToString(arr2[0]));  
+ return (Object[][]) ArrayUtils.addAll(arr1, arr2);  
+ }  
========================================

FALCON-563 Fix checkstyle issues of falcon-regression contributed by Raghav Kumar Gautam  
  
Project: falcon.txt  
Commit: ad51ca03f3af9fe7dcec47291c9f7a6ea586cb1d  
**ClassName: org.apache.falcon.regression.core.util.AssertUtil.java  
@@ -199,3 +199,3 @@ public static void assertFailedWithStatus(final ServiceResponse response, final** final String message) throws JAXBException {  
**- Assert.assertNotEquals(response.message, "null", "response message should not be null");  
+ Assert.assertNotEquals(response.getMessage(), "null", "response message should not be null");  
 Assert.assertEquals(Util.parseResponse(response).getStatus(),**========================================

refactor Parser - create explicit Tokenizer to enable fix for:  
FELIX-1487: commands on multiple lines - NEWLINE can now be used instead of semiclon as a statement separator.  
The Parser now tracks the exact location (line and column) when an error occurs.  
also fixes:  
FELIX-1473: allow method calls on Strings  
FELIX-1493: add $argv for access to List rather than expanded $args  
FELIX-1474: result of commands is implicitly written to pipe  
  
Project: felix.txt  
Commit: c578f444bb3c2bc192ee74294398fdfac8a237a9  
ClassName: org.apache.felix.gogo.runtime.shell.Tokenizer.java  
  
+ public Type next()  
+ {  
+ final Type prevType = type;  
+ token = null;  
+ value = null;  
+ short tLine;  
+ short tColumn;  
+ while (true)  
+ {  
+ skipSpace();  
+ tLine = line;  
+ tColumn = column;  
+ switch (ch)  
+ {  
+ case EOT:  
+ type = Type.EOT;  
+ break;  
+ case '\n':  
+ getch();  
+ if (inArray)  
+ continue;  
+ switch (prevType)  
+ {  
+ case PIPE:  
+ case SEMICOLON:  
+ case NEWLINE:  
+ continue;  
+ default:  
+ type = Type.NEWLINE;  
+ break;  
+ }  
+ break;  
+ case '{':  
+ case '(':  
+ case '[':  
+ value = group();  
+ getch();  
+ break;  
+ case ';':  
+ getch();  
+ type = Type.SEMICOLON;  
+ break;  
+ case '|':  
+ getch();  
+ type = Type.PIPE;  
+ break;  
+ case '=':  
+ if (firstWord || inArray)  
+ {  
+ getch();  
+ type = Type.ASSIGN;  
+ break;  
+ }  
+ default:  
+ value = word();  
+ type = Type.WORD;  
+ }  
+ firstWord = (Type.WORD == type && (Type.WORD != prevType && Type.ASSIGN != prevType));  
+ token = new Token(type, value, tLine, tColumn);  
+ if (DEBUG)  
+ {  
+ System.err.print("<" + type + ">");  
+ if (Type.EOT == type)  
+ {  
+ System.err.println();  
+ }  
+ }  
+ return type;  
+ }  
+ }  
+ private CharSequence word()  
+ {  
+ int start = index - 1;  
+ int skipCR = 0;  
+ do  
+ {  
+ switch (ch)  
+ {  
+ case '\n':  
+ if (index >= 2 && text.charAt(index - 2) == '\r')  
+ skipCR = 1;  
+ case '=':  
+ if ((Type.WORD == type || Type.ASSIGN == type) && '=' == ch  
+ && !inArray)  
+ continue;  
+ case ' ':  
+ case '\t':  
+ case '|':  
+ case ';':  
+ return text.subSequence(start, index - 1 - skipCR);  
+ case '{':  
+ group();  
+ break;  
+ case '\\':  
+ escape();  
+ break;  
+ case '\'':  
+ case '"':  
+ skipQuote();  
+ break;  
+ }  
+ }  
+ while (getch() != EOT);  
+ return text.subSequence(start, index - 1);  
+ }  
+ private CharSequence group()  
+ {  
+ final char push = ch;  
+ final char pop;  
+ switch (ch)  
+ {  
+ case '{':  
+ type = Type.CLOSURE;  
+ pop = '}';  
+ break;  
+ case '(':  
+ type = Type.EXECUTION;  
+ pop = ')';  
+ break;  
+ case '[':  
+ type = Type.ARRAY;  
+ pop = ']';  
+ break;  
+ default:  
**+ assert false;**+ pop = 0;  
+ }  
+ short sLine = line;  
+ short sCol = column;  
+ int start = index;  
+ int depth = 1;  
+ while (true)  
+ {  
+ boolean comment = false;  
+ switch (ch)  
+ {  
+ case '{':  
+ case '(':  
+ case '[':  
+ case '\n':  
+ comment = true;  
+ break;  
+ }  
+ if (getch() == EOT)  
+ {  
+ throw new EOFError(sLine, sCol, "unexpected EOT looking for matching '"  
+ + pop + "'");  
+ }  
+ if (comment || isBlank(ch))  
+ skipSpace();  
+ switch (ch)  
+ {  
+ case '"':  
+ case '\'':  
+ skipQuote();  
+ break;  
+ case '\\':  
+ ch = escape();  
+ break;  
+ default:  
+ if (push == ch)  
+ depth++;  
+ else if (pop == ch && --depth == 0)  
+ return text.subSequence(start, index - 1);  
+ }  
+ }  
+ }  
+ private char escape()  
+ {  
**+ assert '\\' == ch;**+ switch (getch())  
+ {  
+ case 'u':  
+ getch();  
+ getch();  
+ getch();  
+ getch();  
+ if (EOT == ch)  
+ {  
+ throw new EOFError(line, column, "unexpected EOT in \\u escape");  
+ }  
+ String u = text.subSequence(index - 4, index).toString();  
+ try  
+ {  
+ return (char) Integer.parseInt(u, 16);  
+ }  
+ catch (NumberFormatException e)  
+ {  
+ throw new SyntaxError(line, column, "bad unicode escape: \\u" + u);  
+ }  
+ case EOT:  
+ throw new EOFError(line, column, "unexpected EOT in \\ escape");  
+ case '\n':  
+ return '\0'; // line continuation  
+ case '\\':  
+ case '\'':  
+ case '"':  
+ case '$':  
+ return ch;  
+ default:  
+ return ch;  
+ }  
+ }  
+ private void skipQuote()  
+ {  
**+ assert '\'' == ch || '"' == ch;**+ final char quote = ch;  
+ final short sLine = line;  
+ final short sCol = column;  
+ while (getch() != EOT)  
+ {  
+ if (quote == ch)  
+ return;  
+ if ((quote == '"') && ('\\' == ch))  
+ escape();  
+ }  
+ throw new EOFError(sLine, sCol, "unexpected EOT looking for matching quote: "  
+ + quote);  
+ }  
+ private void skipSpace()  
+ {  
+ while (true)  
+ {  
+ while (isBlank(ch))  
+ {  
+ getch();  
+ }  
+ if (('\\' == ch) && (peek() == '\n'))  
+ {  
+ getch();  
+ getch();  
+ continue;  
+ }  
+ if (('/' == ch) || ('#' == ch))  
+ {  
+ if (('#' == ch) || (peek() == '/'))  
+ {  
+ while ((getch() != EOT) && ('\n' != ch))  
+ {  
+ }  
+ continue;  
+ }  
+ else if ('\*' == peek())  
+ {  
+ short sLine = line;  
+ short sCol = column;  
+ getch();  
+ while ((getch() != EOT) && !(('\*' == ch) && (peek() == '/')))  
+ {  
+ }  
+ if (EOT == ch)  
+ {  
+ throw new EOFError(sLine, sCol,  
+ "unexpected EOT looking for closing comment: \*/");  
+ }  
+ getch();  
+ getch();  
+ continue;  
+ }  
+ }  
+ break;  
+ }  
+ }  
+ private boolean isBlank(char ch)  
+ {  
+ return ' ' == ch || '\t' == ch;  
+ }  
+ private boolean isName(char ch)  
+ {  
+ return Character.isJavaIdentifierPart(ch) && (ch != '$') || ('.' == ch);  
+ }  
+ public static Object expand(CharSequence word, Evaluate eval)  
+ {  
+ return expand(word, eval, false);  
+ }  
+ private static Object expand(CharSequence word, Evaluate eval,  
+ boolean inQuote)  
+ {  
+ final String special = "$\\\"'";  
+ int i = word.length();  
+ while ((--i >= 0) && (special.indexOf(word.charAt(i)) == -1))  
+ {  
+ }  
+ if (i < 0)  
+ return word;  
+ return new Tokenizer(word, eval, inQuote).expand();  
+ }  
+ public Object expand(CharSequence word, short line, short column)  
+ {  
+ return expand(new Token(Type.WORD, word, line, column), evaluate, inQuote);  
+ }  
+ private Token word(CharSequence value)  
+ {  
+ return new Token(Type.WORD, value, line, column);  
+ }  
+ private Object expand()  
+ {  
+ StringBuilder buf = new StringBuilder();  
+ while (ch != EOT)  
+ {  
+ int start = index;  
+ switch (ch)  
+ {  
+ case '$':  
+ Object val = expandVar();  
+ if (EOT == ch && buf.length() == 0)  
+ {  
+ return val;  
+ }  
+ if (null != val)  
+ {  
+ buf.append(val);  
+ }  
+ continue; // expandVar() has already read next char  
+ case '\\':  
+ ch = (inQuote && ("u$\\\n\"".indexOf(peek()) == -1)) ? '\\'  
+ : escape();  
+ if (ch != '\0') // ignore line continuation  
+ {  
+ buf.append(ch);  
+ }  
+ break;  
+ case '"':  
+ Token ww = word(null);  
+ skipQuote();  
+ ww.value = text.subSequence(start, index - 1);  
+ value = ww;  
+ Object expand = expand(value, evaluate, true);  
+ if (eot() && buf.length() == 0 && value.equals(expand))  
+ {  
+ return ww.value;  
+ }  
+ if (null != expand)  
+ {  
+ buf.append(expand.toString());  
+ }  
+ break;  
+ case '\'':  
+ if (!inQuote)  
+ {  
+ skipQuote();  
+ value = text.subSequence(start, index - 1);  
+ if (eot() && buf.length() == 0)  
+ {  
+ return value;  
+ }  
+ buf.append(value);  
+ break;  
+ }  
+ default:  
+ buf.append(ch);  
+ }  
+ getch();  
+ }  
+ return buf.toString();  
+ }  
+ private Object expandVar()  
+ {  
**+ assert '$' == ch;**+ Object val;  
+ if (getch() != '{')  
+ {  
+ int start = index - 1;  
+ while (isName(ch))  
+ {  
+ getch();  
+ }  
+ if (index - 1 == start)  
+ {  
+ val = "$";  
+ }  
+ else  
+ {  
+ String name = text.subSequence(start, index - 1).toString();  
+ val = evaluate.get(name);  
+ }  
+ }  
+ else  
+ {  
+ short sLine = line;  
+ short sCol = column;  
+ CharSequence group = group();  
+ char c;  
+ int i = 0;  
+ while (i < group.length())  
+ {  
+ switch (group.charAt(i))  
+ {  
+ case ':':  
+ case '-':  
+ case '+':  
+ case '=':  
+ case '?':  
+ break;  
+ default:  
+ ++i;  
+ continue;  
+ }  
+ break;  
+ }  
+ sCol += i;  
+ String name = String.valueOf(expand(group.subSequence(0, i), sLine, sCol));  
+ for (int j = 0; j < name.length(); ++j)  
+ {  
+ if (!isName(name.charAt(j)))  
+ {  
+ throw new SyntaxError(sLine, sCol, "bad name: ${" + group + "}");  
+ }  
+ }  
+ val = evaluate.get(name);  
+ if (i < group.length())  
+ {  
+ c = group.charAt(i++);  
+ if (':' == c)  
+ {  
+ c = (i < group.length() ? group.charAt(i++) : EOT);  
+ }  
+ CharSequence word = group.subSequence(i, group.length());  
+ switch (c)  
+ {  
+ case '-':  
+ case '=':  
+ if (null == val)  
+ {  
+ val = expand(word, evaluate, false);  
+ if ('=' == c)  
+ {  
+ evaluate.put(name, val);  
+ }  
+ }  
+ break;  
+ case '+':  
+ if (null != val)  
+ {  
+ val = expand(word, evaluate, false);  
+ }  
+ break;  
+ case '?':  
+ if (null == val)  
+ {  
+ val = expand(word, evaluate, false);  
+ if (null == val || val.toString().length() == 0)  
+ {  
+ val = "parameter not set";  
+ }  
+ throw new IllegalArgumentException(name + ": " + val);  
+ }  
+ break;  
+ default:  
+ throw new SyntaxError(sLine, sCol, "bad substitution: ${" + group  
+ + "}");  
+ }  
+ }  
+ getch();  
+ }  
+ return val;  
+ }  
+ private boolean eot()  
+ {  
+ return index >= text.length();  
+ }  
+ private char getch()  
+ {  
+ return ch = getch(false);  
+ }  
+ private char peek()  
+ {  
+ return getch(true);  
+ }  
+ private char getch(boolean peek)  
+ {  
+ if (eot())  
+ {  
+ if (!peek)  
+ {  
+ ++index;  
+ ch = EOT;  
+ }  
+ return EOT;  
+ }  
+ int current = index;  
+ char c = text.charAt(index++);  
+ if (('\r' == c) && !eot() && (text.charAt(index) == '\n'))  
+ c = text.charAt(index++);  
+ if (peek)  
+ {  
+ index = current;  
+ }  
+ else if ('\n' == c)  
+ {  
+ ++line;  
+ column = 0;  
+ }  
+ else  
+ ++column;  
+ return c;  
+ }  
+}  
========================================

fixed serialization error, still issues in restarting  
  
Project: flink.txt  
Commit: 27fa26b163b00fc842eb74b15d603a26504ac949  
ClassName: eu.stratosphere.pact.runtime.hash.HashMatchIteratorITCase.java  
  
+ public static void afterClass() {  
+ if (ioManager != null) {  
+ ioManager.shutdown();  
+ if (!ioManager.isProperlyShutDown()) {  
**+ Assert.fail("I/O manager failed to properly shut down.");**+ }  
+ ioManager = null;  
+ }  
+ }  
========================================

fixed serialization error, still issues in restarting  
  
Project: flink.txt  
Commit: 27fa26b163b00fc842eb74b15d603a26504ac949  
ClassName: eu.stratosphere.pact.runtime.hash.MultiLevelHashITCase.java  
  
+ private void checkBoundaries(BucketBoundaries[] boundaries) {  
+ for (int level = 0; level < boundaries.length; level++) {  
+ int lowerBound = boundaries[level].getLowerBound();  
+ int upperBound = boundaries[level].getUpperBound();  
+ int bucketCountInLevel = 0;  
+ int bucketCountOutOfRange = 0;  
+ SortedMap<Integer, Integer> levelMap = bucketSizesPerLevel  
+ .get(level);  
+ Iterator<Integer> bucketSizeIterator = levelMap.keySet().iterator();  
+ while (bucketSizeIterator.hasNext()) {  
+ int bucketSize = bucketSizeIterator.next();  
+ if (bucketSize != 0) {  
+ int countForBucketSize = levelMap.get(bucketSize);  
+ bucketCountInLevel += countForBucketSize;  
+ if (lowerBound > bucketSize || upperBound < bucketSize) {  
+ bucketCountOutOfRange += countForBucketSize;  
+ }  
+ }  
+ }  
+ double bucketsOutOfRange = (double) bucketCountOutOfRange  
+ / (double) bucketCountInLevel;  
+ double maxBucketsOutOfRange = boundaries[level]  
+ .getPercentOutOfRange();  
**+ Assert.assertTrue("More than " + (maxBucketsOutOfRange \* 100)**+ + "% of buckets out of range in level " + level,  
+ bucketsOutOfRange <= maxBucketsOutOfRange);  
+ int maxEmpty = boundaries[level].getMaxEmpty();  
**+ Assert.assertTrue(**+ "More than " + maxEmpty + " empty buckets in level "  
+ + level,  
+ (maxEmpty == BucketBoundaries.MAX\_EMPTY\_UNBOUNDED)  
+ || (levelMap.get(0) <= boundaries[level]  
+ .getMaxEmpty()));  
+ }  
+ }  
========================================

o Applied patch GERONIMO-11 (TYPO error of Ordinal word (oridnal))  
by Emerson Cargnin  
PR:  
Obtained from:  
Submitted by:  
Reviewed by:  
  
Project: geronimo.txt  
Commit: f6bad98d8f767db2e586149b11cde9809b530195  
ClassName: org.apache.geronimo.remoting.InvocationSupport.java  
  
@@ -94,3 +94,3 @@ public static void putInvocationType(Invocation invocation, InvocationType type)  
 private InvocationSupport(String name, int ordinal) {  
**- assert ordinal < MAX\_ORIDNAL;  
+ assert ordinal < MAX\_ORDINAL;  
 assert values[ordinal] == null;**========================================

o RMIClassLoaderSpi (and test) to get around a problem with spaces in  
codebase strings. See issue GERONIMO-22  
PR:  
Obtained from:  
Submitted by:  
Reviewed by:  
  
Project: geronimo.txt  
Commit: c3bbd6f91618dde93bf20f8c36275f3a400ab3cf  
ClassName: org.apache.geronimo.rmi.RMIClassLoaderSpiImpl.java  
  
+ static URL normalizeURL(URL url)  
+ {  
**+ assert url != null;**+ if (url.getProtocol().equals("file")) {  
+ String filename = url.getFile().replace('/', File.separatorChar);  
+ File file = new File(filename);  
+ try {  
+ url = file.toURI().toURL();  
+ }  
+ catch (MalformedURLException ignore) {}  
+ }  
+ return url;  
+ }  
========================================

o Applied patch GERONIMO-11 (TYPO error of Ordinal word (oridnal))  
by Emerson Cargnin  
PR:  
Obtained from:  
Submitted by:  
Reviewed by:  
  
Project: geronimo.txt  
Commit: f6bad98d8f767db2e586149b11cde9809b530195  
ClassName: org.apache.geronimo.common.InvocationType.java  
  
@@ -91,3 +91,3 @@ public static void putType(Invocation invocation, InvocationType type) {  
 private InvocationType(String name, int ordinal, boolean local, boolean home) {  
**- assert ordinal < MAX\_ORIDNAL;  
+ assert ordinal < MAX\_ORDINAL;  
 assert values[ordinal] == null;**========================================

o Applied deploy-url-\* patches from bug GERONIMO-22 (spaces in deploy url)  
by Brent Worden  
\* This only fixes part of the problem as far as I can tell  
PR:  
Obtained from:  
Submitted by:  
Reviewed by:  
  
Project: geronimo.txt  
Commit: 7c5d2e562f20eca1b46c1b91b4cdce7cbb927390  
ClassName: org.apache.geronimo.deployment.goal.DeployURL.java  
  
+ private static URL normalize(URL url) {  
**+ assert url != null;**+ File f = toFile(url);  
+ if (f != null) {  
+ try {  
+ url = f.toURI().toURL();  
+ }  
+ catch (MalformedURLException ignore) {  
+ }  
+ }  
========================================

o Applied patch GERONIMO-11 (TYPO error of Ordinal word (oridnal))  
by Emerson Cargnin  
PR:  
Obtained from:  
Submitted by:  
Reviewed by:  
  
Project: geronimo.txt  
Commit: f6bad98d8f767db2e586149b11cde9809b530195  
ClassName: org.apache.geronimo.ejb.container.EJBContainerUtil.java  
  
@@ -124,3 +124,3 @@ public static void putCredentials(Invocation invocation, Object credentials) {  
 private EJBContainerUtil(String name, int ordinal) {  
**- assert ordinal < MAX\_ORIDNAL;  
+ assert ordinal < MAX\_ORDINAL;  
 assert values[ordinal] == null;**========================================

o Applied patch GERONIMO-11 (TYPO error of Ordinal word (oridnal))  
by Emerson Cargnin  
PR:  
Obtained from:  
Submitted by:  
Reviewed by:  
  
Project: geronimo.txt  
Commit: f6bad98d8f767db2e586149b11cde9809b530195  
ClassName: org.apache.geronimo.ejb.EJBInvocationUtil.java  
  
@@ -133,3 +133,3 @@ public static void putCredentials(Invocation invocation, Object credentials) {  
 private EJBInvocationUtil(String name, int ordinal) {  
**- assert ordinal < MAX\_ORIDNAL;  
+ assert ordinal < MAX\_ORDINAL;  
 assert values[ordinal] == null;**========================================

Addded synchronization code to fix multi threaded state change issues.  
Changed relationship code to assume relationship type is always  
registered as the relation service does not send notifications when  
a relation type is added or removed.  
  
Project: geronimo.txt  
Commit: 40cf70816d34bf1ed8ffac9ed94360bb1ca26b22  
ClassName: org.apache.geronimo.common.AbstractStateManageable.java  
  
 public final void start() throws Exception {  
**+ assert !Thread.holdsLock(this): "This method cannot be called while holding a syncrhonized lock on this";**+ synchronized (this) {  
 State state = getStateInstance();  
========================================

Addded synchronization code to fix multi threaded state change issues.  
Changed relationship code to assume relationship type is always  
registered as the relation service does not send notifications when  
a relation type is added or removed.  
  
Project: geronimo.txt  
Commit: 40cf70816d34bf1ed8ffac9ed94360bb1ca26b22  
ClassName: org.apache.geronimo.common.AbstractStateManageable.java  
  
- private final void doNotification(String s) {  
- sendNotification(new Notification(s, this, sequenceNumber++));  
+ private final void doNotification(String type) {  
**+ assert !Thread.holdsLock(this): "This method cannot be called while holding a syncrhonized lock on this";**+ sendNotification(new Notification(type, this, sequenceNumber++));  
 }  
========================================

Addded synchronization code to fix multi threaded state change issues.  
Changed relationship code to assume relationship type is always  
registered as the relation service does not send notifications when  
a relation type is added or removed.  
  
Project: geronimo.txt  
Commit: 40cf70816d34bf1ed8ffac9ed94360bb1ca26b22  
ClassName: org.apache.geronimo.common.AbstractStateManageable.java  
  
+ protected void checkState() {  
**+ assert !Thread.holdsLock(this): "This method cannot be called while holding a syncrhonized lock on this";**+ State newState = null;  
+ try {  
+ synchronized (this) {  
 State state = getStateInstance();  
 if (state == State.STARTING) {  
- if (dependencyService.canStart(objectName)) {  
+ if (dependencyService.canStart(objectName) && canStart()) {  
 try {  
========================================

Addded synchronization code to fix multi threaded state change issues.  
Changed relationship code to assume relationship type is always  
registered as the relation service does not send notifications when  
a relation type is added or removed.  
  
Project: geronimo.txt  
Commit: 40cf70816d34bf1ed8ffac9ed94360bb1ca26b22  
ClassName: org.apache.geronimo.common.AbstractStateManageable.java  
  
 public final void startRecursive() throws Exception {  
**+ assert !Thread.holdsLock(this): "This method cannot be called while holding a syncrhonized lock on this";** State state = getStateInstance();  
- if (state == State.STOPPING) {  
- throw new IllegalArgumentException("Cannot startRecursive while in the stopping state");  
+ if (state != State.STOPPED && state != State.FAILED) {  
+ return;  
 }  
========================================

Improved error message for MojoSupport hooks  
Adding tools:show-dependencies goal, which is a helper to render a tree of the projects dependencies  
Add some support for dependency listing lifted from the maven-project-info-plugin  
  
Project: geronimo-genesis.txt  
Commit: f4133ba6658cc312cc13451e78f3152e1b09b3ba  
ClassName: org.apache.geronimo.genesis.DependencyResolutionListener.java  
  
+ public void omitForNearer(Artifact omitted, Artifact kept) {  
**+ assert omitted.getDependencyConflictId().equals(kept.getDependencyConflictId());**+ Node prev = (Node) artifacts.get(omitted.getDependencyConflictId());  
+ if (prev != null) {  
+ if (prev.parent != null) {  
+ prev.parent.children.remove(prev);  
+ }  
+ artifacts.remove(omitted.getDependencyConflictId());  
+ }  
+ includeArtifact(kept);  
+ }  
========================================

Improve debugging and error handling of scriptpath  
  
Project: geronimo-genesis.txt  
Commit: deac3b000dbc5384336e2a0341dd40fcd8864bdb  
ClassName: org.apache.geronimo.genesis.plugins.script.GroovyMojo.java  
  
@@ -147,7 +163,4 @@ public URL loadGroovySource(final String classname) throws MalformedURLException  
 for (int i=0; i<scriptpath.length; i++) {  
- if (scriptpath[i] == null) {  
- log.warn("Ignoring null element in scriptpath");  
- }  
- else {  
**+ assert scriptpath[i] != null;** File file = new File(scriptpath[i], resource);  
========================================

Improved error message for MojoSupport hooks  
Adding tools:show-dependencies goal, which is a helper to render a tree of the projects dependencies  
Add some support for dependency listing lifted from the maven-project-info-plugin  
  
Project: geronimo-genesis.txt  
Commit: f4133ba6658cc312cc13451e78f3152e1b09b3ba  
ClassName: org.apache.geronimo.genesis.DependencyResolutionListener.java  
  
+ public void endProcessChildren(Artifact artifact) {  
+ Node check = (Node) parents.pop();  
**+ assert artifact.equals(check.artifact);**+ currentDepth--;  
+ }  
========================================

Hooked up the security bits, fixed up the handshake+login messages... and well, its soo close I can feel it  
  
Project: geronimo-gshell.txt  
Commit: b216e35b4843b78f3834c60a7e08abbd5dedba30  
ClassName: org.apache.geronimo.gshell.remote.transport.tcp.TcpTransportSupport.java  
  
+ public void setMessageCodecFactory(final MessageCodecFactory codecFactory) {  
**+ assert codecFactory != null;**+ log.debug("Using codec factory: {}", codecFactory);  
========================================

Added sleep command  
Fixed bug in exit  
  
Project: geronimo-gshell.txt  
Commit: b9de761d444b94a50bf781bba2434a16bda90323  
ClassName: org.apache.geronimo.gshell.commands.standard.SleepCommand.java  
  
+ protected int doExecute(String[] args) throws Exception {  
**+ assert args != null;**+ IO io = getIO();  
+ Options options = new Options();  
+ options.addOption(OptionBuilder.withLongOpt("help")  
+ .withDescription("Display this help message")  
+ .create('h'));  
+ CommandLineParser parser = new PosixParser();  
+ CommandLine line = parser.parse(options, args);  
+ args = line.getArgs();  
+ boolean usage = false;  
+ long time = -1;  
+ if (args.length != 1) {  
+ usage = true;  
+ }  
+ else {  
+ time = Long.parseLong(args[0]);  
+ }  
+ if (usage || line.hasOption('h')) {  
+ io.out.println(getName() + " -- sleep... zzzZ");  
+ io.out.println();  
+ HelpFormatter formatter = new HelpFormatter();  
+ formatter.printHelp(  
+ io.out,  
+ 80, // width (FIXME: Should pull from gshell.columns variable)  
+ getName() + " [options] <time>",  
+ "",  
+ options,  
+ 4, // left pad  
+ 4, // desc pad  
+ "",  
+ false); // auto usage  
+ io.out.println();  
+ return Command.SUCCESS;  
+ }  
+ log.info("Sleeping for " + time);  
+ try {  
+ Thread.sleep(time);  
+ }  
+ catch (InterruptedException ignore) {  
+ log.debug("Sleep was interrupted... :-(");  
+ }  
+ log.info("Awake now");  
+ return Command.SUCCESS;  
+ }  
========================================

More bits to get the remote shell crapo working, can execute remote commands now, and the result is passed back, as well as any exceptions as faults  
Still having some problem getting the IO briding crapo to work, some simple commands like "echo foo" will work, and write back to the client's IO now...  
but it will freakout after a while... not sure why... blah  
Have to give each remoteshell its own fresh container to avoid pulling in unwanted components from the invoking shell  
Have to invoke remoteshell's execute() bits in a seperate thread to prevent the message system from blocking up  
And other stuff that is related  
  
Project: geronimo-gshell.txt  
Commit: a4147e334dc70445bf2f1636dc66c5849d226495  
ClassName: org.apache.geronimo.gshell.remote.client.RshClient.java  
  
- public Object execute(final String line) throws Exception {  
**- assert line != null;**+ private Object doExecute(final ExecuteMessage msg) throws Exception {  
**+ assert msg != null;**- log.info("Executing: {}", line);  
+ ExecuteMessage.Result result = (ExecuteMessage.Result) transport.request(msg);  
- transport.send(new ExecuteMessage(line));  
+ if (result instanceof ExecuteMessage.Fault) {  
+ ExecuteMessage.Fault fault = (ExecuteMessage.Fault)result;  
- return null;  
+ throw new Exception("Remote command execution fault", fault.getCause());  
 }  
- public Object execute(final Object... args) throws Exception {  
**- assert args != null;**+ Object rv = result.getResult();  
- log.info("Executing: {}", args);  
+ log.info("Command result: {}", rv);  
- transport.send(new ExecuteMessage(args));  
+ return rv;  
+ }  
+ public Object execute(final String line) throws Exception {  
**+ assert line != null;**+ log.info("Executing (String): {}", line);  
- return null;  
+ return doExecute(new ExecuteMessage(line));  
 }  
========================================

(GSHELL-116) Dropped the need for IOLookup, EnvironmentLookup and TerminalFactory bits, will have some affect on the rsh stuff, which I'll fix up later  
  
Project: geronimo-gshell.txt  
Commit: 2961e57af15fc468bb564021ba482d7ba24d7bc2  
ClassName: org.apache.geronimo.gshell.branding.DefaultBranding.java  
  
 public String getVersion() {  
**+ assert versionLoader != null;** return versionLoader.getVersion();  
========================================

Hooked up the security bits, fixed up the handshake+login messages... and well, its soo close I can feel it  
  
Project: geronimo-gshell.txt  
Commit: b216e35b4843b78f3834c60a7e08abbd5dedba30  
ClassName: org.apache.geronimo.gshell.remote.transport.tcp.TcpTransportSupport.java  
  
 public void setProtocolHandler(final TcpProtocolHandler protocolHandler) {  
**+ assert protocolHandler != null;** log.debug("Using protocol handler: {}", protocolHandler);  
========================================

More bits to get the remote shell crapo working, can execute remote commands now, and the result is passed back, as well as any exceptions as faults  
Still having some problem getting the IO briding crapo to work, some simple commands like "echo foo" will work, and write back to the client's IO now...  
but it will freakout after a while... not sure why... blah  
Have to give each remoteshell its own fresh container to avoid pulling in unwanted components from the invoking shell  
Have to invoke remoteshell's execute() bits in a seperate thread to prevent the message system from blocking up  
And other stuff that is related  
  
Project: geronimo-gshell.txt  
Commit: a4147e334dc70445bf2f1636dc66c5849d226495  
ClassName: org.apache.geronimo.gshell.remote.message.MessageSupport.java  
  
+ protected void writeObject(final ByteBuffer out, final Object obj) throws IOException {  
**+ assert out != null;**+ byte[] bytes = null;  
+ if (obj != null) {  
+ ByteArrayOutputStream baos = new ByteArrayOutputStream();  
+ ObjectOutputStream oos = new ObjectOutputStream(baos);  
+ oos.writeObject(obj);  
+ oos.flush();  
+ bytes = baos.toByteArray();  
+ }  
+ writeBytes(out, bytes);  
+ }  
========================================

More bits to get the remote shell crapo working, can execute remote commands now, and the result is passed back, as well as any exceptions as faults  
Still having some problem getting the IO briding crapo to work, some simple commands like "echo foo" will work, and write back to the client's IO now...  
but it will freakout after a while... not sure why... blah  
Have to give each remoteshell its own fresh container to avoid pulling in unwanted components from the invoking shell  
Have to invoke remoteshell's execute() bits in a seperate thread to prevent the message system from blocking up  
And other stuff that is related  
  
Project: geronimo-gshell.txt  
Commit: a4147e334dc70445bf2f1636dc66c5849d226495  
ClassName: org.apache.geronimo.gshell.remote.message.MessageSupport.java  
  
+ protected void writeBuffer(final ByteBuffer out, final ByteBuffer buffer) {  
**+ assert out != null;**+ if (buffer == null) {  
+ writeBytes(out, null);  
+ }  
+ else {  
+ writeBoolean(out, false);  
+ out.putInt(buffer.remaining());  
+ out.put(buffer);  
+ }  
+ }  
========================================

Hooked up the security bits, fixed up the handshake+login messages... and well, its soo close I can feel it  
  
Project: geronimo-gshell.txt  
Commit: b216e35b4843b78f3834c60a7e08abbd5dedba30  
ClassName: org.apache.geronimo.gshell.remote.message.CryptoAwareMessageSupport.java  
  
+ protected void encryptString(final ByteBuffer out, final Key key, final String str) throws Exception {  
**+ assert out != null;  
+ assert key != null;**+ if (str == null) {  
+ out.putInt(-1);  
+ }  
+ else {  
+ byte[] bytes = getCryptoContext().encrypt(key, str.getBytes());  
+ out.putInt(bytes.length);  
+ out.put(bytes);  
+ }  
+ }  
========================================

Tidy up and note about throwing exceptions, also fix order thrown  
  
Project: geronimo-gshell.txt  
Commit: c75b959892f0432b3530b95e8723f96e007559c7  
ClassName: org.apache.geronimo.gshell.DefaultCommandExecutor.java  
  
 public Object execute(final Object[][] commands) throws Exception {  
**+ assert commands != null;**========================================

More bits to get the remote shell crapo working, can execute remote commands now, and the result is passed back, as well as any exceptions as faults  
Still having some problem getting the IO briding crapo to work, some simple commands like "echo foo" will work, and write back to the client's IO now...  
but it will freakout after a while... not sure why... blah  
Have to give each remoteshell its own fresh container to avoid pulling in unwanted components from the invoking shell  
Have to invoke remoteshell's execute() bits in a seperate thread to prevent the message system from blocking up  
And other stuff that is related  
  
Project: geronimo-gshell.txt  
Commit: a4147e334dc70445bf2f1636dc66c5849d226495  
ClassName: org.apache.geronimo.gshell.remote.message.MessageSupport.java  
  
+ protected boolean readBoolean(final ByteBuffer in) {  
**+ assert in != null;**+ byte b = in.get();  
+ if (b == TRUE) {  
+ return true;  
+ }  
========================================

HDFS-2709. Appropriately handle error conditions in EditLogTailer. Contributed by Aaron T. Myers.  
  
Project: hadoop.txt  
Commit: 9a07ba8945407cd8f63169faf9e0faa4311d38c7  
ClassName: org.apache.hadoop.hdfs.server.namenode.EditLogFileInputStream.java  
  
+ public void skipTransactions(long transactionsToSkip) throws IOException {  
**+ assert firstTxId != HdfsConstants.INVALID\_TXID &&**+ lastTxId != HdfsConstants.INVALID\_TXID;  
+ for (long i = 0; i < transactionsToSkip; i++) {  
+ reader.readOp();  
+ }  
+ }  
========================================

HDFS-2692. Fix bugs related to failover from/into safe mode. Contributed by Todd Lipcon.  
  
Project: hadoop.txt  
Commit: 20a6560bdfd8c4e3b6c3ac9b0f0f62d32e3a2191  
ClassName: org.apache.hadoop.hdfs.server.namenode.FSNamesystem.java  
  
- private boolean isConsistent() {  
+ private void doConsistencyCheck() {  
**+ boolean assertsOn = false;  
+ assert assertsOn = true; // set to true if asserts are on  
+ if (!assertsOn) return;**+ int activeBlocks = blockManager.getActiveBlockCount();  
 if (blockTotal == -1 && blockSafe == -1) {  
- return true; // manual safe mode  
+ return; // manual safe mode  
+ }  
========================================

HDFS-2693. Fix synchronization issues around state transition. Contributed by Todd Lipcon.  
  
Project: hadoop.txt  
Commit: 36d1c49486587c2dbb193e8538b1d4510c462fa6  
ClassName: org.apache.hadoop.hdfs.server.blockmanagement.BlockManager.java  
  
@@ -819,4 +819,3 @@ public void findAndMarkBlockAsCorrupt(final ExtendedBlock blk,  
 final DatanodeInfo dn) throws IOException {  
- namesystem.writeLock();  
- try {  
**+ assert namesystem.hasWriteLock();** final BlockInfo storedBlock = getStoredBlock(blk.getLocalBlock());  
========================================

HDFS-2692. Fix bugs related to failover from/into safe mode. Contributed by Todd Lipcon.  
  
Project: hadoop.txt  
Commit: 20a6560bdfd8c4e3b6c3ac9b0f0f62d32e3a2191  
ClassName: org.apache.hadoop.hdfs.server.namenode.FSNamesystem.java  
  
 private void checkMode() {  
**+ assert hasWriteLock();** if (needEnter()) {  
========================================

HBASE-8101 Cleanup: findbugs and javadoc warning fixes as well as making it illegal passing null row to Put/Delete, etc.  
  
Project: hbase.txt  
Commit: 466902ffde5087c45377cc2f8badb3ff0740cf23  
ClassName: org.apache.hadoop.hbase.io.encoding.DiffKeyDeltaEncoder.java  
  
@@ -211,4 +211,5 @@ private void uncompressSingleKeyValue(DataInputStream source,  
 state.familyNameWithSize[0] = state.familyLength;  
- source.read(state.familyNameWithSize, KeyValue.FAMILY\_LENGTH\_SIZE,  
+ int read = source.read(state.familyNameWithSize, KeyValue.FAMILY\_LENGTH\_SIZE,  
 state.familyLength);  
**+ assert read == state.familyLength;** }  
========================================

HBASE-7443 More findbugs fixes  
  
Project: hbase.txt  
Commit: 92c98a76d111c4f8870bd8be15f3cff6fd0ff51b  
ClassName: org.apache.hadoop.hbase.security.access.AccessController.java  
  
@@ -851,4 +849,6 @@ public void preGetClosestRowBefore(final ObserverContext<RegionCoprocessorEnviro  
 throws IOException {  
**+ assert family != null;** requirePermission("getClosestRowBefore", Permission.Action.READ, c.getEnvironment(),  
- (family != null ? Lists.newArrayList(family) : null));  
+ Lists.newArrayList(family));  
 }  
========================================

HBASE-9862 manage error per server and per region in the protobuffed client  
  
Project: hbase.txt  
Commit: 7f9e701bb6f2d54675fdbe9d62862c7994334662  
ClassName: org.apache.hadoop.hbase.client.AsyncProcess.java  
  
- private void receiveMultiAction(List<Action<Row>> initialActions,  
- MultiAction<Row> rsActions, HRegionLocation location,  
+ private void receiveMultiAction(List<Action<Row>> initialActions, MultiAction<Row> multiAction,  
+ HRegionLocation location,  
 MultiResponse responses, int numAttempt,  
 HConnectionManager.ServerErrorTracker errorsByServer) {  
- if (responses == null) {  
- LOG.info("#" + id + ", attempt #" + numAttempt + "/" + numTries +  
- " failed all ops, trying resubmit," + location);  
- resubmitAll(initialActions, rsActions, location, numAttempt + 1, null, errorsByServer);  
- return;  
- }  
**+ assert responses != null;**========================================

HBASE-5521 [jira] Move compression/decompression to an encoder specific encoding  
context  
Author: Yongqiang He  
Summary:  
https://issues.apache.org/jira/browse/HBASE-5521  
As part of working on HBASE-5313, we want to add a new columnar encoder/decoder.  
It makes sense to move compression to be part of encoder/decoder:  
1) a scanner for a columnar encoded block can do lazy decompression to a  
specific part of a key value object  
2) avoid an extra bytes copy from encoder to hblock-writer.  
If there is no encoder specified for a writer, the HBlock.Writer will use a  
default compression-context to do something very similar to today's code.  
Test Plan: existing unit tests verified by mbautin and tedyu. And no new test  
added here since this code is just a preparation for columnar encoder. Will add  
testcase later in that diff.  
Reviewers: dhruba, tedyu, sc, mbautin  
Reviewed By: mbautin  
Differential Revision: https://reviews.facebook.net/D2097  
  
Project: hbase.txt  
Commit: 6be64072e85dd0341560448c5c710411b46c8a70  
ClassName: org.apache.hadoop.hbase.io.hfile.HFileBlock.java  
  
@@ -1810,53 +1670,35 @@ private HFileBlock readBlockDataInternal(FSDataInputStream is, long offset,  
 throw new IOException("Failed to read compressed block at "  
- + offset + ", onDiskSizeWithoutHeader=" + onDiskSizeWithHeader  
- + ", preReadHeaderSize=" + preReadHeaderSize  
- + ", header.length=" + header.length + ", header bytes: "  
- + Bytes.toStringBinary(header, 0, hdrSize), ex);  
- }  
+ + offset  
+ + ", onDiskSizeWithoutHeader="  
+ + onDiskSizeWithHeader  
+ + ", preReadHeaderSize="  
+ + hdrSize  
+ + ", header.length="  
+ + prefetchedHeader.header.length  
+ + ", header bytes: "  
+ + Bytes.toStringBinary(prefetchedHeader.header, 0,  
+ hdrSize), ex);  
+ }  
+ int onDiskSizeWithoutHeader = onDiskSizeWithHeader - hdrSize;  
**+ assert onDiskSizeWithoutHeader >= 0;** b.validateOnDiskSizeWithoutHeader(onDiskSizeWithoutHeader);  
- b.nextBlockOnDiskSizeWithHeader = nextBlockOnDiskSize;  
- if (verifyChecksum &&   
- !validateBlockChecksum(b, onDiskBlock, hdrSize)) {  
- return null; // checksum mismatch  
- }  
- DataInputStream dis = new DataInputStream(new ByteArrayInputStream(  
- onDiskBlock, hdrSize, onDiskSizeWithoutHeader));  
- b.allocateBuffer(b.nextBlockOnDiskSizeWithHeader > 0);  
- decompress(b.buf.array(), b.buf.arrayOffset() + hdrSize, dis,  
- b.uncompressedSizeWithoutHeader);  
- if (nextBlockOnDiskSize > 0) {  
- System.arraycopy(onDiskBlock, onDiskSizeWithHeader, b.buf.array(),  
- b.buf.arrayOffset() + hdrSize  
- + b.uncompressedSizeWithoutHeader + b.totalChecksumBytes(),   
- hdrSize);  
- setNextBlockHeader(offset, b);  
- }  
- }  
 } else {  
- PrefetchedHeader prefetchedHeader = prefetchedHeaderForThread.get();  
- ByteBuffer headerBuf = prefetchedHeader.offset == offset ?  
- prefetchedHeader.buf : null;  
- if (headerBuf == null) {  
+ if (headerBuf == null) {  
 headerBuf = ByteBuffer.allocate(hdrSize);  
- readAtOffset(is, headerBuf.array(), headerBuf.arrayOffset(), hdrSize,  
- false, offset, pread);  
+ readAtOffset(is, headerBuf.array(), headerBuf.arrayOffset(),  
+ hdrSize, false, offset, pread);  
 }  
========================================

fix tests  
  
Project: helix.txt  
Commit: 3e44bdaab666bf4a46a9c6fa7403bf8ed0cc3033  
ClassName: com.linkedin.clustermanager.zk.ZkStandAloneCMHandler.java  
  
@@ -216,3 +218,8 @@ protected void stopCurrentLeader(String clusterName)  
 logger.info("stop current leader:" + controller);  
 **Assert.assertTrue(controller != null);**+ ClusterManager manager = \_managerMap.remove(controller);  
**+ Assert.assertTrue(manager != null);**+ manager.disconnect();  
 Thread thread = \_threadMap.remove(controller);  
========================================

HIVE-2642 fix Hive-2566 and make union optimization more aggressive  
(Yongqiang He via namit)  
  
Project: hive.txt  
Commit: f8476afc00f75ba0a625c575e94eb5217851e9c6  
ClassName: org.apache.hadoop.hive.ql.optimizer.GenMapRedUtils.java  
  
@@ -342,4 +352,48 @@ public static void initUnionPlan(GenMRProcContext opProcCtx,  
 new PartitionDesc(tt\_desc, null));  
- plan.getAliasToWork().put(taskTmpDir, currUnionOp);  
+ plan.getAliasToWork().put(taskTmpDir, topOperators.get(pos));  
+ }  
+ }  
+ }  
+ }  
+ }  
+ public static void initUnionPlan(GenMRProcContext opProcCtx,  
+ Task<? extends Serializable> currTask, boolean local)  
+ throws SemanticException {  
+ MapredWork plan = (MapredWork) currTask.getWork();  
+ UnionOperator currUnionOp = opProcCtx.getCurrUnionOp();  
**+ assert currUnionOp != null;**+ GenMRUnionCtx uCtx = opProcCtx.getUnionTask(currUnionOp);  
**+ assert uCtx != null;**+ setUnionPlan(opProcCtx, local, plan, uCtx, false);  
 }  
========================================

HIVE-635. fix union accepting multiple objectinspectors  
(Zheng Shao via namit)  
  
Project: hive.txt  
Commit: dc3dfeafd30787ae03553164ffcdfdce1ede25b8  
ClassName: org.apache.hadoop.hive.ql.exec.UnionOperator.java  
  
- public void initializeOp(Configuration hconf, Reporter reporter, ObjectInspector[] inputObjInspector) throws HiveException {  
- initializeChildren(hconf, reporter, inputObjInspector);  
+ public void initializeOp(Configuration hconf, Reporter reporter,  
+ ObjectInspector[] inputObjInspector) throws HiveException {  
+ int parents = inputObjInspector.length;  
+ parentObjInspectors = new StructObjectInspector[parents];  
+ parentFields = new List[parents];  
+ for (int p = 0; p < parents; p++) {  
+ parentObjInspectors[p] = (StructObjectInspector)inputObjInspector[p];  
+ parentFields[p] = parentObjInspectors[p].getAllStructFieldRefs();  
+ }  
+ int columns = parentFields[0].size();  
+ ArrayList<String> columnNames = new ArrayList<String>(columns);  
+ for (int c = 0; c < columns; c++) {  
+ columnNames.add(parentFields[0].get(c).getFieldName());  
+ }  
+ columnTypeResolvers = new ReturnObjectInspectorResolver[columns];  
+ for (int c = 0; c < columns; c++) {  
+ columnTypeResolvers[c] = new ReturnObjectInspectorResolver();  
+ }  
+ for (int p = 0; p < parents; p++) {  
**+ assert(parentFields[p].size() == columns);**+ for (int c = 0; c < columns; c++) {  
+ columnTypeResolvers[c].update(parentFields[p].get(c).getFieldObjectInspector());  
+ }  
+ }  
+ ArrayList<ObjectInspector> outputFieldOIs = new ArrayList<ObjectInspector>(columns);  
+ for (int c = 0; c < columns; c++) {  
+ outputFieldOIs.add(columnTypeResolvers[c].get());  
+ }  
+ outputOI = ObjectInspectorFactory.getStandardStructObjectInspector(  
+ columnNames, outputFieldOIs);  
+ outputRow = new ArrayList<Object>(columns);  
+ for (int c = 0; c < columns; c++) {  
+ outputRow.add(null);  
+ }  
+ needsTransform = new boolean[parents];  
+ for (int p = 0; p < parents; p++) {  
+ needsTransform[p] = (inputObjInspector[p] != outputOI);  
+ if (needsTransform[p]) {  
+ LOG.info("Union Operator needs to transform row from parent[" + p + "] from "  
+ + inputObjInspector[p] + " to " + outputOI);  
+ }  
+ }  
+ initializeChildren(hconf, reporter, new ObjectInspector[] {outputOI});  
 }  
========================================

HIVE-1134. Bucketed Map Join bug fix  
(He Yongqiang via namit)  
M CHANGES.txt  
A ql/src/test/results/clientpositive/bucketmapjoin\_negative.q.out  
M ql/src/test/results/clientpositive/bucketmapjoin2.q.out  
A ql/src/test/results/clientpositive/bucketmapjoin5.q.out  
M ql/src/test/results/clientpositive/bucketmapjoin3.q.out  
M ql/src/test/results/clientpositive/bucketmapjoin1.q.out  
A ql/src/test/results/clientpositive/bucketmapjoin\_negative2.q.out  
M ql/src/test/results/clientpositive/bucketmapjoin4.q.out  
M ql/src/test/queries/clientpositive/bucketmapjoin1.q  
A ql/src/test/queries/clientpositive/bucketmapjoin\_negative.q  
M ql/src/test/queries/clientpositive/bucketmapjoin2.q  
M ql/src/test/queries/clientpositive/bucketmapjoin3.q  
A ql/src/test/queries/clientpositive/bucketmapjoin\_negative2.q  
M ql/src/test/queries/clientpositive/bucketmapjoin4.q  
A ql/src/test/queries/clientpositive/bucketmapjoin5.q  
M ql/src/java/org/apache/hadoop/hive/ql/optimizer/GenMapRedUtils.java  
M ql/src/java/org/apache/hadoop/hive/ql/optimizer/BucketMapJoinOptimizer.java  
M ql/src/java/org/apache/hadoop/hive/ql/exec/ExecMapper.java  
M ql/src/java/org/apache/hadoop/hive/ql/plan/MapredLocalWork.java  
M ql/src/java/org/apache/hadoop/hive/ql/plan/MapredWork.java  
  
Project: hive.txt  
Commit: ed81bc5ddefde8f0363e104823f57627a4f19d94  
ClassName: org.apache.hadoop.hive.ql.optimizer.BucketMapJoinOptimizer.java  
  
@@ -281,14 +329,52 @@ public Object process(Node nd, Stack<Node> stack, NodeProcessorCtx procCtx,  
 continue;  
- Collections.sort(aliasToBucketFileNames.get(alias));  
+ Collections.sort(aliasToBucketFileNamesMapping.get(alias));  
 LinkedHashMap<String, ArrayList<String>> mapping = new LinkedHashMap<String, ArrayList<String>>();  
 aliasBucketFileNameMapping.put(alias, mapping);  
- for(String inputBigTBLBucket : aliasToBucketFileNames.get(baseBigAlias)) {  
- int smallTblBucketNum = aliasToBucketNumber.get(alias);  
+ if (bigTblPartsToBucketNumber.size() > 0) {  
+ Iterator<Entry<Partition, List<String>>> bigTblPartToBucketNames = bigTblPartsToBucketFileNames  
+ .entrySet().iterator();  
+ Iterator<Entry<Partition, Integer>> bigTblPartToBucketNum = bigTblPartsToBucketNumber  
+ .entrySet().iterator();  
+ while (bigTblPartToBucketNames.hasNext()) {  
**+ assert bigTblPartToBucketNum.hasNext();**+ int bigTblBucketNum = bigTblPartToBucketNum.next().getValue().intValue();  
+ List<String> bigTblBucketNameList = bigTblPartToBucketNames.next().getValue();  
+ fillMapping(baseBigAlias, aliasToBucketNumberMapping,  
+ aliasToBucketFileNamesMapping, alias, mapping, bigTblBucketNum,  
+ bigTblBucketNameList);  
+ }  
+ } else {  
+ List<String> bigTblBucketNameList = aliasToBucketFileNamesMapping.get(baseBigAlias);  
+ int bigTblBucketNum = aliasToBucketNumberMapping.get(baseBigAlias);  
+ fillMapping(baseBigAlias, aliasToBucketNumberMapping,  
+ aliasToBucketFileNamesMapping, alias, mapping, bigTblBucketNum,  
+ bigTblBucketNameList);  
+ }  
+ }  
========================================

HIVE-2206 [jira] add a new optimizer for query correlation discovery and optimization  
(Yin Huai via Ashutosh Chauhan)  
Summary:  
update test results  
This issue proposes a new logical optimizer called Correlation Optimizer, which is used to merge correlated MapReduce jobs (MR jobs) into a single MR job. The idea is based on YSmart (http://ysmart.cse.ohio-state.edu/). The paper and slides of YSmart are linked at the bottom.  
Since Hive translates queries in a sentence by sentence fashion, for every operation which may need to shuffle the data (e.g. join and aggregation operations), Hive will generate a MapReduce job for that operation. However, for those operations which may need to shuffle the data, they may involve correlations explained below and thus can be executed in a single MR job.  
Input Correlation: Multiple MR jobs have input correlation (IC) if their input relation sets are not disjoint;  
Transit Correlation: Multiple MR jobs have transit correlation (TC) if they have not only input correlation, but also the same partition key;  
Job Flow Correlation: An MR has job ï¬‚ow correlation (JFC) with one of its child nodes if it has the same partition key as that child node.  
The current implementation of correlation optimizer only detect correlations among MR jobs for reduce-side join operators and reduce-side aggregation operators (not map only aggregation). A query will be optimized if it satisfies following conditions.  
There exists a MR job for reduce-side join operator or reduce side aggregation operator which have JFC with all of its parents MR jobs (TCs will be also exploited if JFC exists);  
All input tables of those correlated MR job are original input tables (not intermediate tables generated by sub-queries); and  
No self join is involved in those correlated MR jobs.  
Correlation optimizer is implemented as a logical optimizer. The main reasons are that it only needs to manipulate the query plan tree and it can leverage the existing component on generating MR jobs.  
Current implementation can serve as a framework for correlation related optimizations. I think that it is better than adding individual optimizers.  
There are several work that can be done in future to improve this optimizer. Here are three examples.  
Support queries only involve TC;  
Support queries in which input tables of correlated MR jobs involves intermediate tables; and  
Optimize queries involving self join.  
References:  
Paper and presentation of YSmart.  
Paper: http://www.cse.ohio-state.edu/hpcs/WWW/HTML/publications/papers/TR-11-7.pdf  
Slides: http://sdrv.ms/UpwJJc  
Test Plan: EMPTY  
Reviewers: JIRA, ashutoshc  
Reviewed By: ashutoshc  
CC: brock  
Differential Revision: https://reviews.facebook.net/D11097  
  
Project: hive.txt  
Commit: b3c86fdda5fcab82a8288a28298f4168d816da84  
ClassName: org.apache.hadoop.hive.ql.optimizer.correlation.IntraQueryCorrelation.java  
  
+ public boolean adjustNumReducers(int newNumReducers) {  
**+ assert newNumReducers != 0;**+ if (newNumReducers > 0) {  
+ if (newNumReducers < minReducers) {  
+ return false;  
+ }  
+ if (numReducers > 0) {  
+ if (newNumReducers != numReducers) {  
+ return false;  
+ }  
+ } else {  
+ numReducers = newNumReducers;  
+ }  
+ }  
+ return true;  
+ }  
========================================

HIVE-2642 fix Hive-2566 and make union optimization more aggressive  
(Yongqiang He via namit)  
  
Project: hive.txt  
Commit: f8476afc00f75ba0a625c575e94eb5217851e9c6  
ClassName: org.apache.hadoop.hive.ql.optimizer.GenMapRedUtils.java  
  
+ public static void joinUnionPlan(GenMRProcContext opProcCtx,  
+ Task<? extends Serializable> currentUnionTask,  
+ Task<? extends Serializable> existingTask, boolean local)  
+ throws SemanticException {  
+ MapredWork plan = (MapredWork) existingTask.getWork();  
+ UnionOperator currUnionOp = opProcCtx.getCurrUnionOp();  
**+ assert currUnionOp != null;**+ GenMRUnionCtx uCtx = opProcCtx.getUnionTask(currUnionOp);  
**+ assert uCtx != null;**+ setUnionPlan(opProcCtx, local, plan, uCtx, true);  
+ List<Task<? extends Serializable>> parTasks = null;  
+ if ((currentUnionTask != null) && (currentUnionTask.getParentTasks() != null)  
+ && !currentUnionTask.getParentTasks().isEmpty()) {  
+ parTasks = new ArrayList<Task<? extends Serializable>>();  
+ parTasks.addAll(currentUnionTask.getParentTasks());  
+ Object[] parTaskArr = parTasks.toArray();  
+ for (Object parTask : parTaskArr) {  
+ ((Task<? extends Serializable>) parTask)  
+ .removeDependentTask(currentUnionTask);  
 }  
========================================

HIVE-2642 fix Hive-2566 and make union optimization more aggressive  
(Yongqiang He via namit)  
  
Project: hive.txt  
Commit: f8476afc00f75ba0a625c575e94eb5217851e9c6  
ClassName: org.apache.hadoop.hive.ql.optimizer.GenMRUnion1.java  
  
+ private void processSubQueryUnionMapJoin(GenMRProcContext ctx) {  
+ AbstractMapJoinOperator<? extends MapJoinDesc> mjOp = ctx.getCurrMapJoinOp();  
**+ assert mjOp != null;**+ GenMRMapJoinCtx mjCtx = ctx.getMapJoinCtx(mjOp);  
**+ assert mjCtx != null;**+ MapredWork plan = (MapredWork) ctx.getCurrTask().getWork();  
+ String taskTmpDir = mjCtx.getTaskTmpDir();  
+ TableDesc tt\_desc = mjCtx.getTTDesc();  
**+ assert plan.getPathToAliases().get(taskTmpDir) == null;**+ plan.getPathToAliases().put(taskTmpDir, new ArrayList<String>());  
+ plan.getPathToAliases().get(taskTmpDir).add(taskTmpDir);  
+ plan.getPathToPartitionInfo().put(taskTmpDir,  
+ new PartitionDesc(tt\_desc, null));  
+ plan.getAliasToWork().put(taskTmpDir, mjCtx.getRootMapJoinOp());  
+ }  
========================================

HIVE-4805 : Enhance coverage of package org.apache.hadoop.hive.ql.exec.errors (Ivan Veselovsky via Ashutosh Chauhan)  
  
Project: hive.txt  
Commit: 496aa313ae601755df21fe3f17688e016615f994  
ClassName: org.apache.hadoop.hive.ql.exec.errors.RegexErrorHeuristic.java  
  
 public void init(String query, JobConf conf) {  
- this.query = query;  
 this.conf = conf;  
**- assert((logRegexes!=null) && (queryRegex != null));  
+ assert(queryRegex != null);**========================================

HIVE-1321. bugs with temp directories, trailing blank fields in HBase bulk load  
(John Sichi via namit)  
  
Project: hive.txt  
Commit: e0287125ac4906b00bc6fea427e25968104bed6f  
ClassName: org.apache.hadoop.hive.hbase.HiveHFileOutputFormat.java  
  
@@ -165,3 +178,3 @@ public void write(Writable w) throws IOException {  
 String [] fields = s.split("\u0001");  
**- assert(fields.length == (columnMap.size() + 1));  
+ assert(fields.length <= (columnMap.size() + 1));**========================================

HIVE-5657 : TopN produces incorrect results with count(distinct) (Sergey Shelukhin via Ashutosh Chauhan)  
  
Project: hive.txt  
Commit: c2ee6682ca22185132b46e73c13b0d437b185c76  
ClassName: org.apache.hadoop.hive.ql.exec.ReduceSinkOperator.java  
  
@@ -245,41 +247,32 @@ public void processOp(Object row, int tag) throws HiveException {  
- for (int i = 0; i < numDistributionKeys; i++) {  
- cachedKeys[0][i] = keyEval[i].evaluate(row);  
- }  
+ populateCachedDistributionKeys(row, 0);  
+ HiveKey firstKey = toHiveKey(cachedKeys[0], tag, null);  
+ int distKeyLength = firstKey.getDistKeyLength();  
 if (numDistinctExprs > 0) {  
- for (int i = 0; i < numDistinctExprs; i++) {  
- if (i > 0) {  
- System.arraycopy(cachedKeys[0], 0, cachedKeys[i], 0, numDistributionKeys);  
- }  
- StandardUnion union = (StandardUnion) cachedKeys[i][numDistributionKeys];  
- if (union == null) {  
- cachedKeys[i][numDistributionKeys] =  
- union = new StandardUnion((byte)i, new Object[distinctColIndices.get(i).size()]);  
- }  
- Object[] distinctParameters = (Object[]) union.getObject();  
- for (int j = 0; j < distinctParameters.length; j++) {  
- distinctParameters[j] =  
- keyEval[distinctColIndices.get(i).get(j)].evaluate(row);  
- }  
- union.setTag((byte) i);  
- }  
+ populateCachedDistinctKeys(row, 0);  
+ firstKey = toHiveKey(cachedKeys[0], tag, distKeyLength);  
+ }  
+ int firstIndex = reducerHash.tryStoreKey(firstKey);  
+ if (firstIndex == TopNHash.EXCLUDE) return; // Nothing to do.  
+ BytesWritable value = makeValueWritable(row);  
+ int hashCode = computeHashCode(row);  
+ if (firstIndex == TopNHash.FORWARD) {  
+ firstKey.setHashCode(hashCode);  
+ collect(firstKey, value);  
+ } else {  
**+ assert firstIndex >= 0;**+ reducerHash.storeValue(firstIndex, value, hashCode, false);  
 }  
- for (int i = 0; i < cachedKeys.length; i++) {  
- Object keyObj = keySerializer.serialize(cachedKeys[i], keyObjectInspector);  
- setKeyWritable(keyIsText ? (Text)keyObj : (BytesWritable)keyObj, tag);  
- int topNIndex = reducerHash.tryStoreKey(keyWritable);  
- if (TopNHash.EXCLUDED == topNIndex) continue;  
- int keyHashCode = computeHashCode(row);  
- BytesWritable valueWritable = getValue(row);  
- if (TopNHash.FORWARD == topNIndex) {  
- keyWritable.setHashCode(keyHashCode);  
- collect(keyWritable, valueWritable);  
- continue;  
- }  
**- assert topNIndex >= 0;**- reducerHash.storeValue(topNIndex, valueWritable, keyHashCode, false);  
+ for (int i = 1; i < numDistinctExprs; i++) {  
+ System.arraycopy(cachedKeys[0], 0, cachedKeys[i], 0, numDistributionKeys);  
+ populateCachedDistinctKeys(row, i);  
+ HiveKey hiveKey = toHiveKey(cachedKeys[i], tag, distKeyLength);  
+ hiveKey.setHashCode(hashCode);  
+ collect(hiveKey, value);  
 }  
========================================

HIVE-2642 fix Hive-2566 and make union optimization more aggressive  
(Yongqiang He via namit)  
  
Project: hive.txt  
Commit: f8476afc00f75ba0a625c575e94eb5217851e9c6  
ClassName: org.apache.hadoop.hive.ql.optimizer.GenMRUnion1.java  
  
+ private void processSubQueryUnionCreateIntermediate(  
+ Operator<? extends Serializable> parent,  
+ Operator<? extends Serializable> child,  
+ Task<? extends Serializable> uTask, GenMRProcContext ctx,  
+ GenMRUnionCtx uCtxTask) {  
+ ParseContext parseCtx = ctx.getParseCtx();  
+ TableDesc tt\_desc = PlanUtils.getIntermediateFileTableDesc(PlanUtils.getFieldSchemasFromRowSchema(  
+ parent.getSchema(), "temporarycol"));  
+ Context baseCtx = parseCtx.getContext();  
+ String taskTmpDir = baseCtx.getMRTmpFileURI();  
+ Operator<? extends Serializable> fs\_op = OperatorFactory.get(  
+ new FileSinkDesc(taskTmpDir, tt\_desc, parseCtx.getConf().getBoolVar(  
+ HiveConf.ConfVars.COMPRESSINTERMEDIATE)), parent.getSchema());  
**+ assert parent.getChildOperators().size() == 1;**+ parent.getChildOperators().set(0, fs\_op);  
+ List<Operator<? extends Serializable>> parentOpList = new ArrayList<Operator<? extends Serializable>>();  
+ parentOpList.add(parent);  
+ fs\_op.setParentOperators(parentOpList);  
+ Operator<? extends Serializable> ts\_op = OperatorFactory.get(  
+ new TableScanDesc(), parent.getSchema());  
+ List<Operator<? extends Serializable>> childOpList = new ArrayList<Operator<? extends Serializable>>();  
+ childOpList.add(child);  
+ ts\_op.setChildOperators(childOpList);  
+ child.replaceParent(parent, ts\_op);  
+ uCtxTask.addTaskTmpDir(taskTmpDir);  
+ uCtxTask.addTTDesc(tt\_desc);  
+ uCtxTask.addListTopOperators(ts\_op);  
+ Task<? extends Serializable> currTask = ctx.getCurrTask();  
+ currTask.addDependentTask(uTask);  
+ if (ctx.getRootTasks().contains(uTask)) {  
+ ctx.getRootTasks().remove(uTask);  
+ if (!ctx.getRootTasks().contains(currTask)) {  
+ ctx.getRootTasks().add(currTask);  
+ }  
+ }  
+ }  
========================================

Initial implementation of the client side request pipelining; many bug fixes in the server side implementation; unit and integration tests  
  
Project: httpcore.txt  
Commit: e8189cdf2845b98aad7e664281d50bcf403315fa  
ClassName: org.apache.http.nio.protocol.HttpAsyncRequestExecutor.java  
  
@@ -138,3 +141,8 @@ public void requestReady(  
 final NHttpClientConnection conn) throws IOException, HttpException {  
- final State state = ensureNotNull(getState(conn));  
+ final State state = getState(conn);  
**+ Asserts.notNull(state, "Connection state");  
+ Asserts.check(state.getRequestState() == MessageState.READY ||**+ state.getRequestState() == MessageState.COMPLETED,  
+ "Unexpected request state %s", state.getRequestState());  
 if (state.getRequestState() != MessageState.READY) {  
========================================

Initial implementation of the client side request pipelining; many bug fixes in the server side implementation; unit and integration tests  
  
Project: httpcore.txt  
Commit: e8189cdf2845b98aad7e664281d50bcf403315fa  
ClassName: org.apache.http.nio.protocol.PipeliningClientExchangeHandler.java  
  
+ public HttpRequest generateRequest() throws IOException, HttpException {  
**+ Asserts.check(this.requestProducer == null, "Inconsistent state: request producer is not null");**+ this.requestProducer = this.requestProducerQueue.poll();  
+ if (this.requestProducer == null) {  
+ return null;  
+ }  
+ final HttpRequest request = this.requestProducer.generateRequest();  
+ this.httppocessor.process(request, this.localContext);  
+ this.requestQueue.add(request);  
+ return request;  
+ }  
========================================

Rewrite of blocking HTTP connection implementations: moved more common code to BHttpConnectionBase, deprecated EntityDeserializer and EntitySerializer, fixed error messages in EncodingUtils  
  
Project: httpcore.txt  
Commit: 969766214dbdbdc3e0f381997476cb5d22096ff9  
ClassName: org.apache.http.impl.BHttpConnectionBase.java  
  
+ protected InputStream getInputStream() {  
**+ Asserts.check(this.instream != null, "Input stream");**+ return this.instream;  
+ }  
========================================

HTTPCORE-24: Fixed bug in non-blocking connection implementations, which prevented the session buffer from being correctly flushed when the content coding process has been completed.  
Contributed by Oleg Kalnichevski  
Reviewed by Asankha C. Perera <asankha at wso2.com>  
  
Project: httpcore.txt  
Commit: 90c02901481cef3490876b73157d174d2926415e  
ClassName: org.apache.http.nio.impl.DefaultNHttpClientConnection.java  
  
@@ -155,2 +156,3 @@ public void submitRequest(final HttpRequest request) throws HttpException {  
 }  
**+ assertNotClosed();**========================================

Initial implementation of the client side request pipelining; many bug fixes in the server side implementation; unit and integration tests  
  
Project: httpcore.txt  
Commit: e8189cdf2845b98aad7e664281d50bcf403315fa  
ClassName: org.apache.http.nio.protocol.PipeliningClientExchangeHandler.java  
  
+ public void responseCompleted() throws IOException {  
**+ Asserts.check(this.responseConsumer != null, "Inconsistent state: response consumer is null");**+ try {  
+ if (!this.keepAlive) {  
+ this.conn.close();  
+ }  
+ this.responseConsumer.responseCompleted(this.localContext);  
+ final T result = this.responseConsumer.getResult();  
+ final Exception ex = this.responseConsumer.getException();  
+ this.responseConsumer = null;  
+ if (result != null) {  
+ this.resultQueue.add(result);  
+ } else {  
+ this.future.failed(ex);  
+ this.conn.shutdown();  
+ }  
+ if (!conn.isOpen()) {  
+ this.done = true;  
+ releaseResources();  
+ }  
+ if (!this.future.isDone() && this.responseConsumerQueue.isEmpty()) {  
+ this.future.completed(new ArrayList<T>(this.resultQueue));  
+ this.resultQueue.clear();  
+ }  
+ } catch (final RuntimeException ex) {  
+ failed(ex);  
+ throw ex;  
+ }  
+ }  
========================================

HTTPCORE-24: Fixed bug in non-blocking connection implementations, which prevented the session buffer from being correctly flushed when the content coding process has been completed.  
Contributed by Oleg Kalnichevski  
Reviewed by Asankha C. Perera <asankha at wso2.com>  
  
Project: httpcore.txt  
Commit: 90c02901481cef3490876b73157d174d2926415e  
ClassName: org.apache.http.nio.impl.DefaultNHttpServerConnection.java  
  
@@ -155,2 +156,3 @@ public void submitResponse(final HttpResponse response) throws HttpException {  
 }  
**+ assertNotClosed();**========================================

HTTPCORE-24: Fixed bug in non-blocking connection implementations, which prevented the session buffer from being correctly flushed when the content coding process has been completed.  
Contributed by Oleg Kalnichevski  
Reviewed by Asankha C. Perera <asankha at wso2.com>  
  
Project: httpcore.txt  
Commit: 90c02901481cef3490876b73157d174d2926415e  
ClassName: org.apache.http.nio.impl.NHttpConnectionBase.java  
  
**+ protected void assertNotClosed() {**+ if (this.closed) {  
+ throw new IllegalStateException("Connection is closed");  
+ }  
========================================

Rewrite of blocking HTTP connection implementations: moved more common code to BHttpConnectionBase, deprecated EntityDeserializer and EntitySerializer, fixed error messages in EncodingUtils  
  
Project: httpcore.txt  
Commit: 969766214dbdbdc3e0f381997476cb5d22096ff9  
ClassName: org.apache.http.impl.BHttpConnectionBase.java  
  
+ protected OutputStream getOutputStream() {  
**+ Asserts.check(this.outstream != null, "Output stream");**+ return this.outstream;  
+ }  
========================================

Initial implementation of the client side request pipelining; many bug fixes in the server side implementation; unit and integration tests  
  
Project: httpcore.txt  
Commit: e8189cdf2845b98aad7e664281d50bcf403315fa  
ClassName: org.apache.http.nio.protocol.PipeliningClientExchangeHandler.java  
  
+ public void requestCompleted() {  
**+ Asserts.check(this.requestProducer != null, "Inconsistent state: request producer is null");**+ this.requestProducer.requestCompleted(this.localContext);  
+ this.requestProducer = null;  
+ }  
========================================

Initial implementation of the client side request pipelining; many bug fixes in the server side implementation; unit and integration tests  
  
Project: httpcore.txt  
Commit: e8189cdf2845b98aad7e664281d50bcf403315fa  
ClassName: org.apache.http.nio.protocol.PipeliningClientExchangeHandler.java  
  
+ public void responseReceived(final HttpResponse response) throws IOException, HttpException {  
**+ Asserts.check(this.responseConsumer == null, "Inconsistent state: response consumer is not null");**+ this.responseConsumer = this.responseConsumerQueue.poll();  
**+ Asserts.check(this.responseConsumer != null, "Inconsistent state: response consumer queue is empty");**+ final HttpRequest request = this.requestQueue.poll();  
**+ Asserts.check(request != null, "Inconsistent state: request queue is empty");**+ this.localContext.setAttribute(HttpCoreContext.HTTP\_REQUEST, request);  
+ this.localContext.setAttribute(HttpCoreContext.HTTP\_RESPONSE, response);  
+ this.httppocessor.process(response, this.localContext);  
+ this.responseConsumer.responseReceived(response);  
+ this.keepAlive = this.connReuseStrategy.keepAlive(response, this.localContext);  
+ }  
========================================

# gg-8123: Isolating the bug (1)  
  
Project: ignite.txt  
Commit: cd94e071139056282d1f2524a8913bae98b69348  
ClassName: org.gridgain.grid.util.offheap.unsafe.GridUnsafeMap0.java  
  
+ private void rehash() {  
+ if (cnt >= MAXIMUM\_CAPACITY || cnt <= threshold)  
+ return;  
+ boolean release = false;  
+ long oldTblAddr = -1;  
+ long oldMemCap = -1;  
+ lock.writeLock().lock();  
+ try {  
+ long oldCap = cap;  
+ oldMemCap = memCap;  
+ oldTblAddr = tblAddr;  
+ if (cnt >= MAXIMUM\_CAPACITY || cnt <= threshold)  
+ return;  
+ long newCap = oldCap << 1;  
+ long newMemCap = newCap \* 8;  
+ long newTblAddr = mem.allocateSystem(newMemCap, true);  
+ long oldTblEnd = oldTblAddr + memCap;  
+ for (long oldBinAddr = oldTblAddr; oldBinAddr < oldTblEnd; oldBinAddr += 8) {  
+ long entryAddr = Bin.first(oldBinAddr, mem);  
+ if (entryAddr == 0)  
+ continue;  
+ while (true) {  
+ int hash = Entry.hash(entryAddr, mem);  
+ long next = Entry.nextAddress(entryAddr, mem);  
+ long newBinAddr = binAddress(hash, newTblAddr, newCap);  
+ long newFirst = Bin.first(newBinAddr, mem);  
+ Bin.first(newBinAddr, entryAddr, mem);  
+ Entry.nextAddress(entryAddr, newFirst, mem);  
+ if (next == 0)  
+ break;  
+ else  
+ entryAddr = next;  
+ }  
+ }  
+ tblAddr = newTblAddr;  
+ memCap = newMemCap;  
+ cap = newCap;  
+ threshold = (long)(newCap \* load);  
+ release = true;  
+ }  
+ finally {  
+ lock.writeLock().unlock();  
+ if (release) {  
**+ assert oldTblAddr != tblAddr;  
+ assert oldTblAddr != -1;  
+ assert oldMemCap != -1;**+ mem.releaseSystem(oldTblAddr, oldMemCap);  
+ }  
+ }  
+ }  
========================================

# Spring cache fixes  
  
Project: ignite.txt  
Commit: 1a2d2ca1e5b008b5dbab1e73513ac61bd9bbcdd9  
ClassName: org.gridgain.grid.cache.spring.GridSpringCacheManager.java  
  
 @Override public void afterPropertiesSet() throws Exception {  
**- assert ignite == null;  
+ assert grid == null;**========================================

# gg-7525: GGFS examples fix.  
  
Project: ignite.txt  
Commit: 2c49586c5acdc34c45617089a21dcfd516843de2  
ClassName: org.gridgain.examples.streaming.StreamingCheckInExample.java  
  
+ public static void main(String[] args) throws Exception {  
+ Timer timer = new Timer("check-in-query-worker");  
+ final Grid g = GridGain.start("examples/config/example-streamer.xml");  
+ try {  
+ GridStreamer streamer = g.streamer(STREAMER\_NAME);  
**+ assert streamer != null;**+ streamer.addStreamerFailureListener(new GridStreamerFailureListener() {  
+ @Override public void onFailure(String stageName, Collection<Object> evts, Throwable err) {  
+ System.err.println("Failure [stage=" + stageName + ", evts=" + evts + ", err=" + err.getMessage());  
+ }  
+ });  
+ scheduleQuery(streamer, timer);  
+ streamData(streamer);  
+ timer.cancel();  
+ g.compute().run(new Runnable() {  
+ @Override public void run() {  
+ GridStreamer streamer = g.streamer(STREAMER\_NAME);  
+ if (streamer == null)  
+ System.err.println("Default streamer not found (is example-streamer.xml " +  
+ "configuration used on all nodes?)");  
+ else {  
+ System.out.println("Clearing streamer data.");  
+ streamer.reset();  
+ }  
+ }  
+ }).get();  
+ }  
+ finally {  
+ GridGain.stop(true);  
+ }  
+ }  
========================================

sp-19 fixed GG-7661  
  
Project: ignite.txt  
Commit: 77f0acd7919cbcde378775cbce42c53fdc0429c6  
ClassName: org.gridgain.grid.kernal.processors.dataload.GridDataLoaderImpl.java  
  
@@ -906,2 +905,6 @@ private void submit(final List<Map.Entry<K, V>> entries, final GridFutureAdapter  
 dep = ctx.deploy().deploy(jobPda0.deployClass(), jobPda0.classLoader());  
+ jobPda0 = jobPda;  
**+ assert jobPda0 != null;** }  
========================================

# Spring cache fixes  
  
Project: ignite.txt  
Commit: 1a2d2ca1e5b008b5dbab1e73513ac61bd9bbcdd9  
ClassName: org.gridgain.grid.cache.spring.GridSpringCacheManager.java  
  
 @Override public Cache getCache(String name) {  
**- assert ignite != null;  
+ assert grid != null;**- return new SpringCache(ignite.cache(name));  
+ try {  
+ return new GridSpringCache(name, grid, grid.cache(name), null);  
+ }  
+ catch (IllegalArgumentException ignored) {  
+ return null;  
+ }  
 }  
========================================

gg-9475 fix review notes  
  
Project: ignite.txt  
Commit: e24613a43885861480eedace4273c838b1fab528  
ClassName: org.gridgain.grid.kernal.processors.hadoop.counter.GridHadoopPerformanceCounter.java  
  
+ public void onJobStart(long ts) {  
**+ assert nodeId != null;**+ evts.add(new T2<>("JOB start " + nodeId, ts));  
+ }  
========================================

#gg-8031 - fixes  
  
Project: ignite.txt  
Commit: 4895e461b29478891aa098b93c5eddbf223ad9a7  
ClassName: org.gridgain.grid.kernal.processors.hadoop.shuffle.collections.GridHadoopSkipList.java  
  
@@ -450,2 +457,4 @@ private void laceUp(Object key, long newMeta, int newMetaLevel) {  
**+ assert cmpRes != 0; // Two different metas with equal keys must be impossible.** break; // Retry cas.  
========================================

# sprint-6 fixed file name  
  
Project: ignite.txt  
Commit: 6759af1ecfcd390b35859e596f9527611ec56d6f  
ClassName: org.apache.ignite.internal.processors.rest.handlers.datastructures.DataStructuresCommandHandler.java  
  
+ @Override public IgniteFuture<GridRestResponse> handleAsync(GridRestRequest req) {  
**+ assert SUPPORTED\_COMMANDS.contains(req.command()) : req.command();**+ return incrementOrDecrement((DataStructuresRequest)req).chain(new CX1<IgniteFuture<?>, GridRestResponse>() {  
+ @Override public GridRestResponse applyx(IgniteFuture<?> fut) throws IgniteCheckedException {  
+ GridRestResponse res = new GridRestResponse();  
+ res.setResponse(fut.get());  
+ return res;  
+ }  
+ });  
+ }  
========================================

GG-6700 - Initial fix draft.  
  
Project: ignite.txt  
Commit: 02bda73d8ea0111aaa3609b9262e042b22cc1d1f  
ClassName: org.gridgain.grid.kernal.processors.cache.distributed.dht.GridDhtLockRequest.java  
  
+ public void markLastKeyForPreload() {  
**+ assert idx > 0;**+ if (preloadKeys == null)  
+ preloadKeys = new BitSet();  
+ preloadKeys.set(idx - 1, true);  
+ }  
+ public boolean needPreloadKey(int idx) {  
+ return preloadKeys != null && preloadKeys.get(idx);  
========================================

GG-8787-1 worked on disco issues  
  
Project: ignite.txt  
Commit: b54d0ca83afd3518c699642099afab512cefc39c  
ClassName: org.gridgain.grid.spi.discovery.tcp.GridTcpDiscoverySpi.java  
  
+ private void prepareNodeAddedMessage(GridTcpDiscoveryAbstractMessage msg) {  
+ if (msg instanceof GridTcpDiscoveryNodeAddedMessage) {  
+ GridTcpDiscoveryNodeAddedMessage nodeAddedMsg =  
+ (GridTcpDiscoveryNodeAddedMessage)msg;  
+ if (nodeAddedMsg.node().equals(next)) {  
+ Collection<GridTcpDiscoveryNode> allNodes = ring.allNodes();  
+ Collection<GridTcpDiscoveryNode> topToSend = new ArrayList<>(allNodes.size());  
+ for (GridTcpDiscoveryNode n0 : allNodes) {  
**+ assert n0.internalOrder() != 0 : n0;**+ if (n0.internalOrder() < nodeAddedMsg.node().internalOrder())  
+ topToSend.add(n0);  
+ }  
+ nodeAddedMsg.topology(topToSend);  
+ Map<Long, Collection<GridNode>> hist;  
+ synchronized (mux) {  
+ hist = new TreeMap<>(topHist);  
+ }  
+ nodeAddedMsg.topologyHistory(hist);  
+ }  
+ }  
+ }  
========================================

StramChild has zero compile errors but it's far from functional. Giving an opportunity for others to synch by committing.  
  
Project: incubator-apex-core.txt  
Commit: e39ecbd0e294ffc7a402c80cd6c1ff2025e9b938  
ClassName: com.malhartech.stram.StramChild.java  
  
- private void deployNode(Kryo kryo, NodeDeployInfo ndi)  
- {  
- Object foreignObject = kryo.readClassAndObject(new Input(ndi.serializedNode));  
- try {  
- Node node = (Node)foreignObject;  
- NodeConfiguration nc = new NodeConfiguration(ndi.properties);  
- node.setup(nc);  
+ Stream stream = streams.get(source);  
+ if (stream == null) {  
**+ assert (nidi.isInline() == false);**+ StreamConfiguration config = new StreamConfiguration();  
+ config.setSocketAddr(StreamConfiguration.SERVER\_ADDRESS, InetSocketAddress.createUnresolved(nidi.bufferServerHost, nidi.bufferServerPort));  
- for ()  
+ stream = new BufferServerInputStream();  
+ stream.setup(config);  
+ Sink s = node.connect(nidi.portName, stream);  
+ stream.connect(ndi.id.concat(".").concat(nidi.portName), s); // what this id should be?  
 }  
- catch (ClassCastException cce) {  
- LOG.error("Expected {} but found {}", Node.class, foreignObject.getClass());  
+ else if (nidi.partitionKeys == null || nidi.partitionKeys.isEmpty()) {  
+ Sink s = node.connect(nidi.portName, stream);  
+ stream.connect(ndi.id.concat(".").concat(nidi.portName), s); // what this id should be?  
+ }  
+ else {  
+ Sink s = node.connect(nidi.portName, stream);  
+ PartitionAwareSink pas = new PartitionAwareSink(new DefaultSerDe(), nidi.partitionKeys, s); // serde should be something else  
+ stream.connect(ndi.id.concat(".").concat(nidi.portName), pas); // what this id should be?  
+ }  
+ streams.put(ndi.id.concat(".").concat(nidi.portName), stream);  
+ }  
+ }  
+ }  
========================================

fix parsing and promotion of int[8|16|32|64] literals  
change ITypePromoteComputer.promote to accept a DataOutput instead of an IMutableValueStorage  
  
Project: incubator-asterixdb.txt  
Commit: a5bdea6d64f4e7d3c614108014790c4cf5f5f8b5  
ClassName: edu.uci.ics.asterix.runtime.operators.file.ADMDataParser.java  
  
+ private void parseAndCast(ATypeTag typeTag, IAType objectType, DataOutput out) throws AsterixException, IOException {  
+ final ATypeTag targetTypeTag = getTargetTypeTag(typeTag, objectType, out);  
+ DataOutput dataOutput = out;  
+ if (targetTypeTag != typeTag) {  
+ castBuffer.reset();  
+ dataOutput = castBuffer.getDataOutput();  
+ }  
+ if (targetTypeTag == null || !parseNumeric(typeTag, dataOutput)) {  
+ throw new AsterixException(mismatchErrorMessage + objectType.getTypeName() + mismatchErrorMessage2  
+ + typeTag);  
+ }  
+ if (targetTypeTag != typeTag) {  
+ ITypePromoteComputer promoteComputer = ATypeHierarchy.getTypePromoteComputer(typeTag, targetTypeTag);  
**+ assert promoteComputer != null;**+ promoteComputer.promote(castBuffer.getByteArray(), castBuffer.getStartOffset() + 1,  
+ castBuffer.getLength() - 1, out);  
 }  
- throw new AsterixException(mismatchErrorMessage + objectType.getTypeName() + mismatchErrorMessage2 + typeTag);  
 }  
========================================

better attempt to fix the bug  
minor cleanup  
  
Project: incubator-asterixdb.txt  
Commit: c191e23414b5d45d2387a9ce2ac22870ea3d6fe7  
ClassName: edu.uci.ics.asterix.transaction.management.service.locking.ConcurrentLockManager.java  
  
**@@ -1055,3 +1063,3 @@ private void assertLocksCanBefound() {** int jobId = jobArenaMgr.getJobId(jobSlot);  
**- assertLockCanBeFound(dsId, entityHashValue, lockMode, jobId);  
+ assertLockCanBeFoundInJobQueue(dsId, entityHashValue, lockMode, jobId);** reqSlot = reqArenaMgr.getNextRequest(reqSlot);  
========================================

a fix and a bunch of debugging changes  
  
Project: incubator-asterixdb.txt  
Commit: c17c9895aa0c3aa2d165ec2cf2012d3f1a0e6da2  
ClassName: edu.uci.ics.asterix.transaction.management.service.locking.ConcurrentLockManager.java  
  
@@ -481,2 +496,5 @@ private void unlock(int dsId, int entityHashValue, byte lockMode, long jobSlot)  
**+ assertLocksCanBefound();** long holder = removeLastHolder(resource, jobSlot, lockMode);  
========================================

1) corrected the way that the logFlusher thread is stopped, 2) fixed the issue of multiple job-level commits, and 3) cleaned up the TransactionContext class  
  
Project: incubator-asterixdb.txt  
Commit: 0e5dacb1f1d596e2f1de06bd7cf0a764b2d740af  
ClassName: edu.uci.ics.asterix.transaction.management.service.transaction.TransactionContext.java  
  
+ public void decrementNumOfActiveJobs() {  
**+ assert numOfActiveJobs > 0;**+ numOfActiveJobs--;  
+ }  
========================================

better attempt to fix the bug  
minor cleanup  
  
Project: incubator-asterixdb.txt  
Commit: c191e23414b5d45d2387a9ce2ac22870ea3d6fe7  
ClassName: edu.uci.ics.asterix.transaction.management.service.locking.ConcurrentLockManager.java  
  
**- private void assertLockCanBeFound(int dsId, int entityHashValue, byte lockMode, int jobId) {**- if (!findLock(dsId, entityHashValue, jobId, lockMode)) {  
**+ private void assertLockCanBeFoundInJobQueue(int dsId, int entityHashValue, byte lockMode, int jobId) {**+ if (findLockInJobQueue(dsId, entityHashValue, jobId, lockMode) == -1) {  
 String msg = "request for " + LockMode.toString(lockMode) + " lock on dataset " + dsId + " entity "  
========================================

better attempt to fix the bug  
minor cleanup  
  
Project: incubator-asterixdb.txt  
Commit: c191e23414b5d45d2387a9ce2ac22870ea3d6fe7  
ClassName: edu.uci.ics.asterix.transaction.management.service.locking.ConcurrentLockManager.java  
  
**- private void assertLocksCanBefound() {  
+ private void assertLocksCanBefoundInJobQueue() throws ACIDException {** for (int i = 0; i < ResourceGroupTable.TABLE\_SIZE; ++i) {  
 final ResourceGroup group = table.table[i];  
- group.getLatch();  
+ group.waitForLatch();  
+ try {  
 long resSlot = group.firstResourceIndex.get();  
========================================

the templates are now a fixed part of the maven plugin  
the package for the generated code is a parameter to the plugin  
  
Project: incubator-asterixdb.txt  
Commit: 28fbd9affc44999382e3e889938f21181212b8a9  
ClassName: main.resources.ArenaManager.java  
  
@@ -64,5 +61,5 @@ public long allocate() {  
**- assert RecordManagerTypes.Global.allocId(result) == allocId;  
- assert RecordManagerTypes.Global.arenaId(result) == localManager.arenaId;  
- assert RecordManagerTypes.Global.localId(result) == localId;  
+ assert TypeUtil.Global.allocId(result) == allocId;  
+ assert TypeUtil.Global.arenaId(result) == localManager.arenaId;  
+ assert TypeUtil.Global.localId(result) == localId;** return result;  
========================================

Introduces Feeds 2.0  
commit c3f577861fc705d848c1641605689cadd6973bae  
Merge: ebc4cae fc0c2c0  
Author: ramangrover29 <ramangrover29@gmail.com>  
Date: Fri Jun 26 13:04:05 2015 -0700  
Merge branch 'raman/feeds\_2\_release' of https://code.google.com/p/asterixdb-sandbox into raman/feeds\_2\_release  
Conflicts:  
asterix-app/src/main/java/edu/uci/ics/asterix/api/http/servlet/FeedServlet.java  
asterix-external-data/src/main/java/edu/uci/ics/asterix/external/library/java/JObjectAccessors.java  
commit ebc4cae21a7302869f953df1ebda601e798d12d2  
Author: ramangrover29 <ramangrover29@gmail.com>  
Date: Sat Jun 20 17:14:45 2015 -0700  
Introduces Feeds 2.0  
Some of the prominent chnages introduced are as follows  
a) Support for building a cascade network of feeds (via secondary feeds feature)  
b) Feed Management Console for tracking active feeds and associated metrics  
c) Support for elastic runtime for data ingestion  
d) Improved fault-tolerance with support for logging of failed records  
Documentation has been added at asterix-doc/src/site/markdown/feeds/  
commit fc0c2c0549a6ee8b202e57607d2e110478cd57bb  
Author: ramangrover29 <ramangrover29@gmail.com>  
Date: Sat Jun 20 17:14:45 2015 -0700  
Introduces Feeds 2.0  
Some of the prominent chnages introduced are as follows  
a) Support for building a cascade network of feeds (via secondary feeds feature)  
b) Feed Management Console for tracking active feeds and associated metrics  
c) Support for elastic runtime for data ingestion  
d) Improved fault-tolerance with support for logging of failed records  
Documentation has been added at asterix-doc/src/site/markdown/feeds/  
Change-Id: I498f01c591a229aaf51cec43ab20f3e5c4f072f4  
Reviewed-on: https://asterix-gerrit.ics.uci.edu/297  
Tested-by: Jenkins <jenkins@fulliautomatix.ics.uci.edu>  
Reviewed-by: Steven Jacobs <sjaco002@ucr.edu>  
Reviewed-by: abdullah alamoudi <bamousaa@gmail.com>  
  
Project: incubator-asterixdb.txt  
Commit: ae85a1dc862c5430095fa9f4a38735a2bfe298f6  
ClassName: edu.uci.ics.asterix.metadata.feeds.FeedLifecycleEventSubscriber.java  
  
**+ public void assertEvent(FeedLifecycleEvent event) throws AsterixException, InterruptedException {**+ boolean eventOccurred = false;  
+ FeedLifecycleEvent e = null;  
+ Iterator<FeedLifecycleEvent> eventsSoFar = inbox.iterator();  
+ while (eventsSoFar.hasNext()) {  
+ e = eventsSoFar.next();  
**+ assertNoFailure(e);**+ eventOccurred = e.equals(event);  
+ }  
+ while (!eventOccurred) {  
+ e = inbox.take();  
+ eventOccurred = e.equals(event);  
+ if (!eventOccurred) {  
**+ assertNoFailure(e);**+ }  
+ }  
+ }  
========================================

a fix and a bunch of debugging changes  
  
Project: incubator-asterixdb.txt  
Commit: c17c9895aa0c3aa2d165ec2cf2012d3f1a0e6da2  
ClassName: edu.uci.ics.asterix.transaction.management.service.locking.ConcurrentLockManager.java  
  
**+ private void assertLockCanBeFound(int dsId, int entityHashValue, byte lockMode, int jobId) {**+ if (!findLock(dsId, entityHashValue, jobId, lockMode)) {  
+ String msg = "request for " + LockMode.toString(lockMode) + " lock on dataset " + dsId + " entity "  
+ + entityHashValue + " not found for job " + jobId + " in thread " + Thread.currentThread().getName();  
+ LOGGER.severe(msg);   
+ throw new IllegalStateException(msg);  
+ }  
+ }  
========================================

check that job is not aborted before requsting another lock  
pass shrinkTimer parameter  
collect some stats  
next free slot is an inaccessible field in each RecordType  
automatically sort fields by size and align items in RecordType  
no more debug fields in RecordType  
skip checks for fields that do not have an intializer in RecordType  
fix RecordManager serialization  
  
Project: incubator-asterixdb.txt  
Commit: f30301111b3f8fb39e8093a302edb186b8bc24b9  
ClassName: edu.uci.ics.asterix.recordmanagergenerator.RecordType.java  
  
+ private void calcOffsetsAndSize() {  
+ Collections.sort(fields, new Comparator<Field>() {  
+ public int compare(Field left, Field right) {  
+ return size(right.type) - size(left.type);  
+ }  
+ });  
+ totalSize = 0;  
+ int alignment = 0;  
+ for (int i = 0; i < fields.size(); ++i) {   
+ final Field field = fields.get(i);  
**+ assert field.offset == -1;**+ field.offset = totalSize;  
+ final int size = size(field.type);  
+ totalSize += size;  
+ if (size > alignment) alignment = size;  
+ }  
+ if (totalSize % alignment != 0) {  
+ totalSize = ((totalSize / alignment) + 1) \* alignment;   
+ }  
 }  
========================================

VariableSizeFrame(VSizeFrame) support for Hyracks.  
This patch replaced Frame/Accessor/Appender with the new API which  
supports BigObject.  
The ExternalSorter/TopKSorter/ExternalGroupSorter  
have been implemented to support big object.  
The Groupby && Join should work with BigObject also. But it will break the  
memory budget when it encounter a big object. I will fix the memory  
problem later in a separate CR.  
The design about the frame allocation is  
here:https://docs.google.com/presentation/d/15h9iQf5OYsgGZoQTbGHkj1yS2G9q2fd0s1lDAD1EJq0/edit?usp=sharing  
Suggest review order:  
Patch 12: It includes all of the sorting operators.  
Patch 13: It applys the new IFrame API to all Hyracks codes.  
Patch 14: Some bug fixes to pass all Asterix's tests.  
Patch 15: Skip it!  
Patch 16: Some bug fixes to the Asterix's tests in small frame setting.  
Later Patch: address the comments  
Change-Id: I2e08692078683f6f2cf17387e39037ad851fc05b  
Reviewed-on: https://asterix-gerrit.ics.uci.edu/234  
Tested-by: Jenkins <jenkins@fulliautomatix.ics.uci.edu>  
Reviewed-by: Yingyi Bu <buyingyi@gmail.com>  
  
Project: incubator-asterixdb-hyracks.txt  
Commit: 0d87a57f7439eca69e6dae73f117747b4ea51746  
ClassName: edu.uci.ics.hyracks.dataflow.std.sort.util.DeletableFrameTupleAppender.java  
  
+ public void reOrganizeBuffer() {  
+ if (deletedSpace <= 0) {  
+ return;  
+ }  
+ reclaimDeletedEnding();  
+ freeDataEndOffset = 0;  
+ int endOffset = 0;  
+ for (int i = 0; i < tupleCount; i++) {  
+ int startOffset = Math.abs(endOffset);  
+ endOffset = getTupleEndOffset(i);  
+ if (endOffset >= 0) {  
+ int length = endOffset - startOffset;  
**+ assert ( length >= 0);**+ if (freeDataEndOffset != startOffset) {  
+ System.arraycopy(array, startOffset, array, freeDataEndOffset, length);  
+ }  
+ freeDataEndOffset += length;  
+ }  
+ setTupleEndOffset(i, freeDataEndOffset);  
+ }  
+ setFreeDataEndOffset(freeDataEndOffset);  
+ deletedSpace = 0;  
+ setDeleteSpace(0);  
+ }  
========================================

VariableSizeFrame(VSizeFrame) support for Hyracks.  
This patch replaced Frame/Accessor/Appender with the new API which  
supports BigObject.  
The ExternalSorter/TopKSorter/ExternalGroupSorter  
have been implemented to support big object.  
The Groupby && Join should work with BigObject also. But it will break the  
memory budget when it encounter a big object. I will fix the memory  
problem later in a separate CR.  
The design about the frame allocation is  
here:https://docs.google.com/presentation/d/15h9iQf5OYsgGZoQTbGHkj1yS2G9q2fd0s1lDAD1EJq0/edit?usp=sharing  
Suggest review order:  
Patch 12: It includes all of the sorting operators.  
Patch 13: It applys the new IFrame API to all Hyracks codes.  
Patch 14: Some bug fixes to pass all Asterix's tests.  
Patch 15: Skip it!  
Patch 16: Some bug fixes to the Asterix's tests in small frame setting.  
Later Patch: address the comments  
Change-Id: I2e08692078683f6f2cf17387e39037ad851fc05b  
Reviewed-on: https://asterix-gerrit.ics.uci.edu/234  
Tested-by: Jenkins <jenkins@fulliautomatix.ics.uci.edu>  
Reviewed-by: Yingyi Bu <buyingyi@gmail.com>  
  
Project: incubator-asterixdb-hyracks.txt  
Commit: 0d87a57f7439eca69e6dae73f117747b4ea51746  
ClassName: edu.uci.ics.hyracks.dataflow.std.util.MathUtil.java  
  
+ public static int log2Floor(int n) {  
**+ assert n >= 1;**+ int log = 0;  
+ if (n > 0xffff) {  
+ n >>>= 16;  
+ log = 16;  
+ }  
+ if (n > 0xff) {  
+ n >>>= 8;  
+ log |= 8;  
+ }  
+ if (n > 0xf) {  
+ n >>>= 4;  
+ log |= 4;  
+ }  
+ if (n > 0b11) {  
+ n >>>= 2;  
+ log |= 2;  
+ }  
+ return log + (n >>> 1);  
+ }  
========================================

added validate() method for IIndexes. Currently only supported for BTree/LSM-BTree  
fixed BTree concurrency bug  
added prettier printing of the BTree  
changed all ordered index tests to validate the tree at the end of each test  
Cleaned up the LSM search cursor and changed LSM BTree search opcallback to  
properly unlatch in-memory pages during reconciliation  
Cleaned up the btree range search cursor  
changed InMemoryBufferCache to also release overflow pages when being reset/closed  
added copytuple method to TupleUtils that generates fewer objects  
  
Project: incubator-asterixdb-hyracks.txt  
Commit: 3b89343c7d02388c836128f65401d133629d761e  
ClassName: edu.uci.ics.hyracks.storage.am.btree.frames.BTreeNSMInteriorFrame.java  
  
+ public void validate(PageValidationInfo pvi) throws HyracksDataException {  
+ int tupleCount = getTupleCount();  
+ for (int i = 0; i < tupleCount; i++) {  
+ frameTuple.resetByTupleIndex(this, i);  
+ if (!pvi.isLowRangeNull) {  
**+ assert cmp.compare(pvi.lowRangeTuple, frameTuple) < 0;**+ }  
+ if (!pvi.isHighRangeNull) {  
**+ assert cmp.compare(pvi.highRangeTuple, frameTuple) >= 0;**+ }  
+ if (i > 0) {  
+ previousFt.resetByTupleIndex(this, i - 1);  
**+ assert cmp.compare(previousFt, frameTuple) < 0;**+ }  
+ }  
========================================

Fixed timing issue between victimization and cleaner. Fixed early termination of clock. Fixed package names. Added HeapBufferAllocator.  
  
Project: incubator-asterixdb-hyracks.txt  
Commit: bc0b3a61157b704d619aac42e56ccf1b5db68f61  
ClassName: edu.uci.ics.hyracks.storage.common.buffercache.BufferCache.java  
  
+ private CachedPage findPage(long dpid, boolean newPage) {  
+ int victimizationTryCount = 0;  
+ int localCleanCount = -1;  
+ synchronized (cleanerThread.cleanNotification) {  
+ localCleanCount = cleanerThread.cleanCount;  
+ }  
+ while (true) {  
+ CachedPage cPage = null;  
+ int hash = hash(dpid);  
+ CacheBucket bucket = pageMap[hash];  
+ bucket.bucketLock.lock();  
+ try {  
+ cPage = bucket.cachedPage;  
+ while (cPage != null) {  
+ if (cPage.dpid == dpid) {  
+ cPage.pinCount.incrementAndGet();  
+ return cPage;  
+ }  
+ cPage = cPage.next;  
+ }  
+ } finally {  
+ bucket.bucketLock.unlock();  
+ }  
+ CachedPage victim = (CachedPage) pageReplacementStrategy.findVictim();  
+ if (victim != null) {  
+ if (victim.dpid < 0) {  
+ bucket.bucketLock.lock();  
+ try {  
+ cPage = bucket.cachedPage;  
+ while (cPage != null) {  
+ if (cPage.dpid == dpid) {  
+ cPage.pinCount.incrementAndGet();  
+ victim.pinCount.decrementAndGet();  
+ return cPage;  
+ }  
+ cPage = cPage.next;  
+ }  
+ victim.reset(dpid);  
+ victim.next = bucket.cachedPage;  
+ bucket.cachedPage = victim;  
+ } finally {  
+ bucket.bucketLock.unlock();  
+ }  
+ return victim;  
+ }  
+ int victimHash = hash(victim.dpid);  
+ if (victimHash == hash) {  
+ bucket.bucketLock.lock();  
+ try {  
+ if (victim.pinCount.get() != 1) {  
+ victim.pinCount.decrementAndGet();  
+ continue;  
+ }  
+ cPage = bucket.cachedPage;  
+ while (cPage != null) {  
+ if (cPage.dpid == dpid) {  
+ cPage.pinCount.incrementAndGet();  
+ victim.pinCount.decrementAndGet();  
+ return cPage;  
+ }  
+ cPage = cPage.next;  
+ }  
+ victim.reset(dpid);  
+ } finally {  
+ bucket.bucketLock.unlock();  
+ }  
+ return victim;  
+ } else {  
+ CacheBucket victimBucket = pageMap[victimHash];  
+ if (victimHash < hash) {  
+ victimBucket.bucketLock.lock();  
+ bucket.bucketLock.lock();  
+ } else {  
+ bucket.bucketLock.lock();  
+ victimBucket.bucketLock.lock();  
+ }  
+ try {  
+ if (victim.pinCount.get() != 1) {  
+ victim.pinCount.decrementAndGet();  
+ continue;  
+ }  
+ cPage = bucket.cachedPage;  
+ while (cPage != null) {  
+ if (cPage.dpid == dpid) {  
+ cPage.pinCount.incrementAndGet();  
+ victim.pinCount.decrementAndGet();  
+ return cPage;  
+ }  
+ cPage = cPage.next;  
+ }  
+ if (victimBucket.cachedPage == victim) {  
+ victimBucket.cachedPage = victim.next;  
+ } else {  
+ CachedPage victimPrev = victimBucket.cachedPage;  
+ while (victimPrev != null && victimPrev.next != victim) {  
+ victimPrev = victimPrev.next;  
+ }  
**+ assert victimPrev != null;**+ victimPrev.next = victim.next;  
+ }  
+ victim.reset(dpid);  
+ victim.next = bucket.cachedPage;  
+ bucket.cachedPage = victim;  
+ } finally {  
+ victimBucket.bucketLock.unlock();  
+ bucket.bucketLock.unlock();  
+ }  
+ return victim;  
+ }  
+ }  
+ if (++victimizationTryCount >= MAX\_VICTIMIZATION\_TRY\_COUNT) {  
+ return null;  
+ }  
+ synchronized (cleanerThread) {  
+ cleanerThread.notifyAll();  
+ }  
+ synchronized (cleanerThread.cleanNotification) {  
+ if (cleanerThread.cleanCount == localCleanCount) {  
+ try {  
+ cleanerThread.cleanNotification.wait(1000);  
+ } catch (InterruptedException e) {  
+ }  
+ }  
+ }  
+ }  
+ }  
========================================

VariableSizeFrame(VSizeFrame) support for Hyracks.  
This patch replaced Frame/Accessor/Appender with the new API which  
supports BigObject.  
The ExternalSorter/TopKSorter/ExternalGroupSorter  
have been implemented to support big object.  
The Groupby && Join should work with BigObject also. But it will break the  
memory budget when it encounter a big object. I will fix the memory  
problem later in a separate CR.  
The design about the frame allocation is  
here:https://docs.google.com/presentation/d/15h9iQf5OYsgGZoQTbGHkj1yS2G9q2fd0s1lDAD1EJq0/edit?usp=sharing  
Suggest review order:  
Patch 12: It includes all of the sorting operators.  
Patch 13: It applys the new IFrame API to all Hyracks codes.  
Patch 14: Some bug fixes to pass all Asterix's tests.  
Patch 15: Skip it!  
Patch 16: Some bug fixes to the Asterix's tests in small frame setting.  
Later Patch: address the comments  
Change-Id: I2e08692078683f6f2cf17387e39037ad851fc05b  
Reviewed-on: https://asterix-gerrit.ics.uci.edu/234  
Tested-by: Jenkins <jenkins@fulliautomatix.ics.uci.edu>  
Reviewed-by: Yingyi Bu <buyingyi@gmail.com>  
  
Project: incubator-asterixdb-hyracks.txt  
Commit: 0d87a57f7439eca69e6dae73f117747b4ea51746  
ClassName: edu.uci.ics.hyracks.dataflow.std.sort.util.GroupVSizeFrame.java  
  
+ private void movePartialFutureToStartPosition() {  
**+ assert buffer.hasArray();**+ if (!FrameHelper.hasBeenCleared(buffer, buffer.position())) {  
+ buffer.compact();  
+ FrameHelper.clearRemainingFrame(buffer, buffer.position()); // mark it to make reset idempotent  
+ }  
+ }  
========================================

fixed btree operator issues causing not serializable exceptions during od shipping  
  
Project: incubator-asterixdb-hyracks.txt  
Commit: 9df9b1547c7e150e3522003cad5b9b2bbb963bd9  
ClassName: edu.uci.ics.hyracks.storage.am.btree.dataflow.BTreeSearchOperatorNodePushable.java  
  
@@ -31,3 +42,13 @@ public void open() throws HyracksDataException {  
 fill();  
- btree.search(cursor, opDesc.getRangePredicate(), leafFrame, interiorFrame);  
**+ assert(searchKeyFields <= btree.getMultiComparator().getKeyLength());**+ IBinaryComparator[] searchComparators = new IBinaryComparator[searchKeyFields];  
+ for(int i = 0; i < searchKeyFields; i++) {  
+ searchComparators[i] = btree.getMultiComparator().getComparators()[i];  
+ }   
+ MultiComparator searchCmp = new MultiComparator(searchComparators, btree.getMultiComparator().getFields());  
+ RangePredicate rangePred = new RangePredicate(isForward, lowKey, highKey, searchCmp);  
+ btree.search(cursor, rangePred, leafFrame, interiorFrame);  
 } catch (FileNotFoundException e1) {  
========================================

Fixed timing issue between victimization and cleaner. Fixed early termination of clock. Fixed package names. Added HeapBufferAllocator.  
  
Project: incubator-asterixdb-hyracks.txt  
Commit: c6b00faee48ea025934921d1bf1d92cbe79bc688  
ClassName: edu.uci.ics.hyracks.storage.common.buffercache.BufferCache.java  
  
+ private CachedPage findPage(long dpid, boolean newPage) {  
+ int victimizationTryCount = 0;  
+ int localCleanCount = -1;  
+ synchronized (cleanerThread.cleanNotification) {  
+ localCleanCount = cleanerThread.cleanCount;  
+ }  
+ while (true) {  
+ CachedPage cPage = null;  
+ int hash = hash(dpid);  
+ CacheBucket bucket = pageMap[hash];  
+ bucket.bucketLock.lock();  
+ try {  
+ cPage = bucket.cachedPage;  
+ while (cPage != null) {  
+ if (cPage.dpid == dpid) {  
+ cPage.pinCount.incrementAndGet();  
+ return cPage;  
+ }  
+ cPage = cPage.next;  
+ }  
+ } finally {  
+ bucket.bucketLock.unlock();  
+ }  
+ CachedPage victim = (CachedPage) pageReplacementStrategy.findVictim();  
+ if (victim != null) {  
+ if (victim.dpid < 0) {  
+ bucket.bucketLock.lock();  
+ try {  
+ cPage = bucket.cachedPage;  
+ while (cPage != null) {  
+ if (cPage.dpid == dpid) {  
+ cPage.pinCount.incrementAndGet();  
+ victim.pinCount.decrementAndGet();  
+ return cPage;  
+ }  
+ cPage = cPage.next;  
+ }  
+ victim.reset(dpid);  
+ victim.next = bucket.cachedPage;  
+ bucket.cachedPage = victim;  
+ } finally {  
+ bucket.bucketLock.unlock();  
+ }  
+ return victim;  
+ }  
+ int victimHash = hash(victim.dpid);  
+ if (victimHash == hash) {  
+ bucket.bucketLock.lock();  
+ try {  
+ if (victim.pinCount.get() != 1) {  
+ victim.pinCount.decrementAndGet();  
+ continue;  
+ }  
+ cPage = bucket.cachedPage;  
+ while (cPage != null) {  
+ if (cPage.dpid == dpid) {  
+ cPage.pinCount.incrementAndGet();  
+ victim.pinCount.decrementAndGet();  
+ return cPage;  
+ }  
+ cPage = cPage.next;  
+ }  
+ victim.reset(dpid);  
+ } finally {  
+ bucket.bucketLock.unlock();  
+ }  
+ return victim;  
+ } else {  
+ CacheBucket victimBucket = pageMap[victimHash];  
+ if (victimHash < hash) {  
+ victimBucket.bucketLock.lock();  
+ bucket.bucketLock.lock();  
+ } else {  
+ bucket.bucketLock.lock();  
+ victimBucket.bucketLock.lock();  
+ }  
+ try {  
+ if (victim.pinCount.get() != 1) {  
+ victim.pinCount.decrementAndGet();  
+ continue;  
+ }  
+ cPage = bucket.cachedPage;  
+ while (cPage != null) {  
+ if (cPage.dpid == dpid) {  
+ cPage.pinCount.incrementAndGet();  
+ victim.pinCount.decrementAndGet();  
+ return cPage;  
+ }  
+ cPage = cPage.next;  
+ }  
+ if (victimBucket.cachedPage == victim) {  
+ victimBucket.cachedPage = victim.next;  
+ } else {  
+ CachedPage victimPrev = victimBucket.cachedPage;  
+ while (victimPrev != null && victimPrev.next != victim) {  
+ victimPrev = victimPrev.next;  
+ }  
**+ assert victimPrev != null;**+ victimPrev.next = victim.next;  
+ }  
+ victim.reset(dpid);  
+ victim.next = bucket.cachedPage;  
+ bucket.cachedPage = victim;  
+ } finally {  
+ victimBucket.bucketLock.unlock();  
+ bucket.bucketLock.unlock();  
+ }  
+ return victim;  
+ }  
+ }  
+ if (++victimizationTryCount >= MAX\_VICTIMIZATION\_TRY\_COUNT) {  
+ return null;  
+ }  
+ synchronized (cleanerThread) {  
+ cleanerThread.notifyAll();  
+ }  
+ synchronized (cleanerThread.cleanNotification) {  
+ if (cleanerThread.cleanCount == localCleanCount) {  
+ try {  
+ cleanerThread.cleanNotification.wait(1000);  
+ } catch (InterruptedException e) {  
+ }  
+ }  
+ }  
+ }  
+ }  
========================================

VariableSizeFrame(VSizeFrame) support for Hyracks.  
This patch replaced Frame/Accessor/Appender with the new API which  
supports BigObject.  
The ExternalSorter/TopKSorter/ExternalGroupSorter  
have been implemented to support big object.  
The Groupby && Join should work with BigObject also. But it will break the  
memory budget when it encounter a big object. I will fix the memory  
problem later in a separate CR.  
The design about the frame allocation is  
here:https://docs.google.com/presentation/d/15h9iQf5OYsgGZoQTbGHkj1yS2G9q2fd0s1lDAD1EJq0/edit?usp=sharing  
Suggest review order:  
Patch 12: It includes all of the sorting operators.  
Patch 13: It applys the new IFrame API to all Hyracks codes.  
Patch 14: Some bug fixes to pass all Asterix's tests.  
Patch 15: Skip it!  
Patch 16: Some bug fixes to the Asterix's tests in small frame setting.  
Later Patch: address the comments  
Change-Id: I2e08692078683f6f2cf17387e39037ad851fc05b  
Reviewed-on: https://asterix-gerrit.ics.uci.edu/234  
Tested-by: Jenkins <jenkins@fulliautomatix.ics.uci.edu>  
Reviewed-by: Yingyi Bu <buyingyi@gmail.com>  
  
Project: incubator-asterixdb-hyracks.txt  
Commit: 0d87a57f7439eca69e6dae73f117747b4ea51746  
ClassName: edu.uci.ics.hyracks.api.comm.FrameHelper.java  
  
+ public static int calcAlignedFrameSizeToStore(int fieldCount, int tupleLength, int minFrameSize) {  
**+ assert fieldCount >= 0 && tupleLength >= 0 && minFrameSize > 0;**+ return (1 + (calcSpaceInFrame(fieldCount, tupleLength) + FrameConstants.META\_DATA\_LEN - 1) / minFrameSize)  
+ }  
========================================

VariableSizeFrame(VSizeFrame) support for Hyracks.  
This patch replaced Frame/Accessor/Appender with the new API which  
supports BigObject.  
The ExternalSorter/TopKSorter/ExternalGroupSorter  
have been implemented to support big object.  
The Groupby && Join should work with BigObject also. But it will break the  
memory budget when it encounter a big object. I will fix the memory  
problem later in a separate CR.  
The design about the frame allocation is  
here:https://docs.google.com/presentation/d/15h9iQf5OYsgGZoQTbGHkj1yS2G9q2fd0s1lDAD1EJq0/edit?usp=sharing  
Suggest review order:  
Patch 12: It includes all of the sorting operators.  
Patch 13: It applys the new IFrame API to all Hyracks codes.  
Patch 14: Some bug fixes to pass all Asterix's tests.  
Patch 15: Skip it!  
Patch 16: Some bug fixes to the Asterix's tests in small frame setting.  
Later Patch: address the comments  
Change-Id: I2e08692078683f6f2cf17387e39037ad851fc05b  
Reviewed-on: https://asterix-gerrit.ics.uci.edu/234  
Tested-by: Jenkins <jenkins@fulliautomatix.ics.uci.edu>  
Reviewed-by: Yingyi Bu <buyingyi@gmail.com>  
  
Project: incubator-asterixdb-hyracks.txt  
Commit: 0d87a57f7439eca69e6dae73f117747b4ea51746  
ClassName: edu.uci.ics.hyracks.dataflow.std.sort.util.GroupFrameAccessor.java  
  
+ private int resetSubTupleAccessor(int tupleIndex) {  
**+ assert tupleIndex < getTupleCount();**+ if (innerFrameInfos.size() == 1) {  
+ return tupleIndex;  
+ }  
+ if (tupleIndex == lastTupleIndex) {  
+ return lastFrameId > 0 ? lastTupleIndex - innerFrameInfos.get(lastFrameId - 1).tupleCount : lastTupleIndex;  
+ }  
+ int subFrameId = Collections.binarySearch(innerFrameInfos, tupleIndex);  
+ if (subFrameId >= 0) {  
+ subFrameId++;  
+ } else {  
+ subFrameId = -subFrameId - 1;  
+ }  
+ frameTupleAccessor.reset(buffer, innerFrameInfos.get(subFrameId).start, innerFrameInfos.get(subFrameId).length);  
+ lastTupleIndex = tupleIndex;  
+ lastFrameId = subFrameId;  
+ return lastFrameId > 0 ? lastTupleIndex - innerFrameInfos.get(lastFrameId - 1).tupleCount : lastTupleIndex;  
+ }  
========================================

VariableSizeFrame(VSizeFrame) support for Hyracks.  
This patch replaced Frame/Accessor/Appender with the new API which  
supports BigObject.  
The ExternalSorter/TopKSorter/ExternalGroupSorter  
have been implemented to support big object.  
The Groupby && Join should work with BigObject also. But it will break the  
memory budget when it encounter a big object. I will fix the memory  
problem later in a separate CR.  
The design about the frame allocation is  
here:https://docs.google.com/presentation/d/15h9iQf5OYsgGZoQTbGHkj1yS2G9q2fd0s1lDAD1EJq0/edit?usp=sharing  
Suggest review order:  
Patch 12: It includes all of the sorting operators.  
Patch 13: It applys the new IFrame API to all Hyracks codes.  
Patch 14: Some bug fixes to pass all Asterix's tests.  
Patch 15: Skip it!  
Patch 16: Some bug fixes to the Asterix's tests in small frame setting.  
Later Patch: address the comments  
Change-Id: I2e08692078683f6f2cf17387e39037ad851fc05b  
Reviewed-on: https://asterix-gerrit.ics.uci.edu/234  
Tested-by: Jenkins <jenkins@fulliautomatix.ics.uci.edu>  
Reviewed-by: Yingyi Bu <buyingyi@gmail.com>  
  
Project: incubator-asterixdb-hyracks.txt  
Commit: 0d87a57f7439eca69e6dae73f117747b4ea51746  
ClassName: edu.uci.ics.hyracks.dataflow.std.sort.buffermanager.VariableTupleMemoryManager.java  
  
+ public boolean insertTuple(IFrameTupleAccessor fta, int idx, TuplePointer tuplePointer)  
+ throws HyracksDataException {  
+ int requiredFreeSpace = calculatePhysicalSpace(fta, idx);  
+ int frameId = findAvailableFrame(requiredFreeSpace);  
+ if (frameId < 0) {  
+ if (canBeInsertedAfterCleanUpFragmentation(requiredFreeSpace)) {  
+ reOrganizeFrames();  
+ frameId = findAvailableFrame(requiredFreeSpace);  
+ statsReOrg++;  
+ } else {  
+ return false;  
+ }  
+ }  
**+ assert frameId >= 0;**+ accessor.reset(frames.get(frameId));  
**+ assert accessor.getContiguousFreeSpace() >= requiredFreeSpace;**+ int tid = accessor.append(fta, idx);  
**+ assert tid >= 0;**+ tuplePointer.reset(frameId, tid);  
+ if (accessor.getContiguousFreeSpace() > MIN\_FREE\_SPACE) {  
+ policy.pushNewFrame(frameId, accessor.getContiguousFreeSpace());  
+ }  
+ numTuples++;  
+ return true;  
+ }  
========================================

Adding a lots fixes to NRT for the blur writer.  
  
Project: incubator-blur.txt  
Commit: 00f93eb42bac57a00630ffba0a84e17ad116d546  
ClassName: com.nearinfinity.blur.manager.writer.nrt.NRTManager.java  
  
+ private boolean waitOnGenCondition(long time, TimeUnit unit)  
+ throws InterruptedException {  
**+ assert reopenLock.isHeldByCurrentThread();**+ if (time < 0) {  
+ newGeneration.await();  
+ return true;  
+ } else {  
+ return newGeneration.await(time, unit);  
+ }  
+ }  
========================================

Adding a lots fixes to NRT for the blur writer.  
  
Project: incubator-blur.txt  
Commit: 00f93eb42bac57a00630ffba0a84e17ad116d546  
ClassName: com.nearinfinity.blur.manager.writer.nrt.SearcherManager.java  
  
+ public void release(IndexSearcher searcher) throws IOException {  
**+ assert searcher != null;**+ searcher.getIndexReader().decRef();  
+ }  
========================================

Adding a lots fixes to NRT for the blur writer.  
  
Project: incubator-blur.txt  
Commit: 00f93eb42bac57a00630ffba0a84e17ad116d546  
ClassName: com.nearinfinity.blur.manager.writer.nrt.NRTManager.java  
  
+ public void close() throws IOException {  
+ reopenLock.lock();  
+ try {  
+ try {  
+ IOUtils.close(withDeletes, withoutDeletes);  
+ } finally { // make sure we signal even if close throws an exception  
+ newGeneration.signalAll();  
+ }  
+ } finally {  
+ reopenLock.unlock();  
**+ assert withDeletes.generation == MAX\_SEARCHER\_GEN && withoutDeletes.generation == MAX\_SEARCHER\_GEN;**+ }  
+ }  
========================================

rename fixing case of Zoo[kK]eeper part 2 of 2  
  
Project: incubator-brooklyn.txt  
Commit: d95dca795076b3f42ff3979c1b952112cdbbc51b  
ClassName: brooklyn.entity.zookeeper.AbstractZooKeeperImpl.java  
  
+ public void waitForServiceUp(long duration, TimeUnit units) {  
+ super.waitForServiceUp(duration, units);  
+ if (((JavaSoftwareProcessDriver)getDriver()).isJmxEnabled()) {  
+ JmxHelper helper = new JmxHelper(this);  
+ try {  
**+ helper.assertMBeanExistsEventually(ZOOKEEPER\_MBEAN, units.toMillis(duration));**+ } finally {  
+ helper.disconnect();  
+ }  
+ }  
+ }  
========================================

SshjTool synchronisation: added TODO and failing test  
- and added a fix for one way that it fails due to the synchronization  
model; but more work required.  
  
Project: incubator-brooklyn.txt  
Commit: de0cf0139d5c58d255f5e1e85324f856f0db0e04  
ClassName: brooklyn.util.internal.ssh.sshj.SshjTool.java  
  
@@ -344,3 +359,17 @@ protected void backoffForAttempt(int retryAttempt, String message) {  
 Stopwatch perfStopwatch = Stopwatch.createStarted();  
- T returnVal = action.create();  
+ T returnVal;  
+ try {  
+ returnVal = action.create();  
**+ } catch (AssertionError e) {**+ throw new IllegalStateException("Problem in "+toString()+" for "+action, e);  
+ }  
 if (LOG.isTraceEnabled()) LOG.trace("<< ({}) acquired {}", toString(), returnVal);  
========================================

better logging for JMX problems, and making Cassandra tests a bit more resilient / informative on errors  
  
Project: incubator-brooklyn.txt  
Commit: f1ae3cbb8923ac14ffb3ab7a815e947982698e02  
ClassName: brooklyn.entity.nosql.cassandra.AstyanaxSupport.java  
  
@@ -63,9 +106,18 @@ public void writeData() throws Exception {  
- AstyanaxContext<Keyspace> context = getAstyanaxContext();  
+ AstyanaxContext<Keyspace> context = getAstyanaxContextForKeyspace("BrooklynIntegrationTest");  
 try {  
 Keyspace keyspace = context.getEntity();  
 try {  
**+ assertNull(keyspace.describeKeyspace().getColumnFamily("Rabbits"));**+ } catch (Exception ek) {  
+ log.debug("repairing Cassandra error by re-creating keyspace "+keyspace+": "+ek);  
+ try {  
+ log.debug("dropping Cassandra keyspace "+keyspace);  
 keyspace.dropKeyspace();  
- } catch (Exception e) { /\* Ignore \*/ }  
+ } catch (Exception e) {  
+ log.debug("Cassandra keyspace "+keyspace+" could not be dropped (probably did not exist): "+e);  
+ }  
+ try {  
 keyspace.createKeyspace(ImmutableMap.<String, Object>builder()  
========================================

18820: Fixed scripts and test cases for descriptive statistics - fixed scripts and test cases for descriptive statistics  
  
Project: incubator-systemml.txt  
Commit: 4bc6e45778bd78aa379715ab2e81fe01db343e19  
ClassName: dml.runtime.matrix.mapred.obsolete.MatrixBlockCRS.java  
  
+ public void incrementalAggregate(AggregateOperator aggOp, MatrixValue newWithCorrection)  
+ throws DMLUnsupportedOperationException, DMLRuntimeException  
+ {  
**+ assert(aggOp.correctionExists);**+ MatrixBlock1D newWithCor=checkType(newWithCorrection);  
+ KahanObject buffer=new KahanObject(0, 0);  
+ if(aggOp.correctionLocation==1)  
+ {  
+ for(int r=0; r<rlen-1; r++)  
+ for(int c=0; c<clen; c++)  
+ {  
+ buffer.\_sum=this.getValue(r, c);  
+ buffer.\_correction=this.getValue(r+1, c);  
+ buffer=(KahanObject) aggOp.increOp.fn.execute(buffer, newWithCor.getValue(r, c),   
+ newWithCor.getValue(r+1, c));  
+ setValue(r, c, buffer.\_sum);  
+ setValue(r+1, c, buffer.\_correction);  
+ }  
+ }else if(aggOp.correctionLocation==2)  
+ {  
+ for(int r=0; r<rlen; r++)  
+ for(int c=0; c<clen-1; c++)  
+ {  
+ buffer.\_sum=this.getValue(r, c);  
+ buffer.\_correction=this.getValue(r, c+1);  
+ buffer=(KahanObject) aggOp.increOp.fn.execute(buffer, newWithCor.getValue(r, c),   
+ newWithCor.getValue(r, c+1));  
+ setValue(r, c, buffer.\_sum);  
+ setValue(r, c+1, buffer.\_correction);  
+ }  
+ }/\*else if(aggOp.correctionLocation==0)  
+ {  
+ for(int r=0; r<rlen; r++)  
+ for(int c=0; c<clen; c++)  
+ {  
+ setValue(r, c, this.getValue(r, c)+newWithCor.getValue(r, c));  
+ }  
+ }\*/else if(aggOp.correctionLocation==3)  
+ {  
+ double n, n2, mu2;  
+ for(int r=0; r<rlen-2; r++)  
+ for(int c=0; c<clen; c++)  
+ {  
+ buffer.\_sum=this.getValue(r, c);  
+ n=this.getValue(r+1, c);  
+ buffer.\_correction=this.getValue(r+2, c);  
+ mu2=newWithCor.getValue(r, c);  
+ n2=newWithCor.getValue(r+1, c);  
+ n=n+n2;  
+ double toadd=(mu2-buffer.\_sum)\*n2/n;  
+ buffer=(KahanObject) aggOp.increOp.fn.execute(buffer, toadd);  
+ setValue(r, c, buffer.\_sum);  
+ setValue(r+1, c, n);  
+ setValue(r+2, c, buffer.\_correction);  
+ }  
+ }else if(aggOp.correctionLocation==4)  
+ {  
+ double n, n2, mu2;  
+ for(int r=0; r<rlen; r++)  
+ for(int c=0; c<clen-2; c++)  
+ {  
+ buffer.\_sum=this.getValue(r, c);  
+ n=this.getValue(r, c+1);  
+ buffer.\_correction=this.getValue(r, c+2);  
+ mu2=newWithCor.getValue(r, c);  
+ n2=newWithCor.getValue(r, c+1);  
+ n=n+n2;  
+ double toadd=(mu2-buffer.\_sum)\*n2/n;  
+ buffer=(KahanObject) aggOp.increOp.fn.execute(buffer, toadd);  
+ setValue(r, c, buffer.\_sum);  
+ setValue(r, c+1, n);  
+ setValue(r, c+2, buffer.\_correction);  
+ }  
+ }  
+ else  
+ throw new DMLRuntimeException("unrecognized correctionLocation: "+aggOp.correctionLocation);  
+ }  
========================================

18820: Fixed scripts and test cases for descriptive statistics - fixed scripts and test cases for descriptive statistics  
  
Project: incubator-systemml.txt  
Commit: 4bc6e45778bd78aa379715ab2e81fe01db343e19  
ClassName: dml.runtime.util.MapReduceTool.java  
  
@@ -465,2 +469,8 @@ public static double pickValue(String dir, NumItemsByEachReducerMetaData metadat  
+ public static int extractNumberFromOutputFile(String name)  
+ {  
+ int i=name.indexOf("part-");  
**+ assert(i>=0);**+ return Integer.parseInt(name.substring(i+5));  
+ }  
========================================

19686: switch to the new sparse matrix format and fixed some junit tests - switch to the new sparse matrix format and fixed some junit tests  
  
Project: incubator-systemml.txt  
Commit: d7ba2e76ecea3c360e332d41e1072b1825af62ff  
ClassName: dml.runtime.matrix.io.MatrixBlockDSM.java  
  
+ public void incrementalAggregate(AggregateOperator aggOp, MatrixValue newWithCorrection)  
+ throws DMLUnsupportedOperationException, DMLRuntimeException  
+ {  
**+ assert(aggOp.correctionExists);**+ MatrixBlockDSM newWithCor=checkType(newWithCorrection);  
+ KahanObject buffer=new KahanObject(0, 0);  
+ if(aggOp.correctionLocation==1)  
+ {  
+ for(int r=0; r<rlen-1; r++)  
+ for(int c=0; c<clen; c++)  
+ {  
+ buffer.\_sum=this.getValue(r, c);  
+ buffer.\_correction=this.getValue(r+1, c);  
+ buffer=(KahanObject) aggOp.increOp.fn.execute(buffer, newWithCor.getValue(r, c),   
+ newWithCor.getValue(r+1, c));  
+ setValue(r, c, buffer.\_sum);  
+ setValue(r+1, c, buffer.\_correction);  
+ }  
+ }else if(aggOp.correctionLocation==2)  
+ {  
+ for(int r=0; r<rlen; r++)  
+ for(int c=0; c<clen-1; c++)  
+ {  
+ buffer.\_sum=this.getValue(r, c);  
+ buffer.\_correction=this.getValue(r, c+1);  
+ buffer=(KahanObject) aggOp.increOp.fn.execute(buffer, newWithCor.getValue(r, c),   
+ newWithCor.getValue(r, c+1));  
+ setValue(r, c, buffer.\_sum);  
+ setValue(r, c+1, buffer.\_correction);  
+ }  
+ }/\*else if(aggOp.correctionLocation==0)  
+ {  
+ for(int r=0; r<rlen; r++)  
+ for(int c=0; c<clen; c++)  
+ {  
+ setValue(r, c, this.getValue(r, c)+newWithCor.getValue(r, c));  
+ }  
+ }\*/else if(aggOp.correctionLocation==3)  
+ {  
+ double n, n2, mu2;  
+ for(int r=0; r<rlen-2; r++)  
+ for(int c=0; c<clen; c++)  
+ {  
+ buffer.\_sum=this.getValue(r, c);  
+ n=this.getValue(r+1, c);  
+ buffer.\_correction=this.getValue(r+2, c);  
+ mu2=newWithCor.getValue(r, c);  
+ n2=newWithCor.getValue(r+1, c);  
+ n=n+n2;  
+ double toadd=(mu2-buffer.\_sum)\*n2/n;  
+ buffer=(KahanObject) aggOp.increOp.fn.execute(buffer, toadd);  
+ setValue(r, c, buffer.\_sum);  
+ setValue(r+1, c, n);  
+ setValue(r+2, c, buffer.\_correction);  
+ }  
+ }else if(aggOp.correctionLocation==4)  
+ {  
+ double n, n2, mu2;  
+ for(int r=0; r<rlen; r++)  
+ for(int c=0; c<clen-2; c++)  
+ {  
+ buffer.\_sum=this.getValue(r, c);  
+ n=this.getValue(r, c+1);  
+ buffer.\_correction=this.getValue(r, c+2);  
+ mu2=newWithCor.getValue(r, c);  
+ n2=newWithCor.getValue(r, c+1);  
+ n=n+n2;  
+ double toadd=(mu2-buffer.\_sum)\*n2/n;  
+ buffer=(KahanObject) aggOp.increOp.fn.execute(buffer, toadd);  
+ setValue(r, c, buffer.\_sum);  
+ setValue(r, c+1, n);  
+ setValue(r, c+2, buffer.\_correction);  
+ }  
+ }  
+ else  
+ throw new DMLRuntimeException("unrecognized correctionLocation: "+aggOp.correctionLocation);  
+ }  
========================================

18820: Fixed scripts and test cases for descriptive statistics - fixed scripts and test cases for descriptive statistics  
  
Project: incubator-systemml.txt  
Commit: 4bc6e45778bd78aa379715ab2e81fe01db343e19  
ClassName: dml.runtime.matrix.mapred.obsolete.MatrixBlockDSM.java  
  
+ public void incrementalAggregate(AggregateOperator aggOp, MatrixValue newWithCorrection)  
+ throws DMLUnsupportedOperationException, DMLRuntimeException  
+ {  
**+ assert(aggOp.correctionExists);**+ MatrixBlockDSM newWithCor=checkType(newWithCorrection);  
+ KahanObject buffer=new KahanObject(0, 0);  
+ if(aggOp.correctionLocation==1)  
+ {  
+ for(int r=0; r<rlen-1; r++)  
+ for(int c=0; c<clen; c++)  
+ {  
+ buffer.\_sum=this.getValue(r, c);  
+ buffer.\_correction=this.getValue(r+1, c);  
+ buffer=(KahanObject) aggOp.increOp.fn.execute(buffer, newWithCor.getValue(r, c),   
+ newWithCor.getValue(r+1, c));  
+ setValue(r, c, buffer.\_sum);  
+ setValue(r+1, c, buffer.\_correction);  
+ }  
+ }else if(aggOp.correctionLocation==2)  
+ {  
+ for(int r=0; r<rlen; r++)  
+ for(int c=0; c<clen-1; c++)  
+ {  
+ buffer.\_sum=this.getValue(r, c);  
+ buffer.\_correction=this.getValue(r, c+1);  
+ buffer=(KahanObject) aggOp.increOp.fn.execute(buffer, newWithCor.getValue(r, c),   
+ newWithCor.getValue(r, c+1));  
+ setValue(r, c, buffer.\_sum);  
+ setValue(r, c+1, buffer.\_correction);  
+ }  
+ }/\*else if(aggOp.correctionLocation==0)  
+ {  
+ for(int r=0; r<rlen; r++)  
+ for(int c=0; c<clen; c++)  
+ {  
+ setValue(r, c, this.getValue(r, c)+newWithCor.getValue(r, c));  
+ }  
+ }\*/else if(aggOp.correctionLocation==3)  
+ {  
+ double n, n2, mu2;  
+ for(int r=0; r<rlen-2; r++)  
+ for(int c=0; c<clen; c++)  
+ {  
+ buffer.\_sum=this.getValue(r, c);  
+ n=this.getValue(r+1, c);  
+ buffer.\_correction=this.getValue(r+2, c);  
+ mu2=newWithCor.getValue(r, c);  
+ n2=newWithCor.getValue(r+1, c);  
+ n=n+n2;  
+ double toadd=(mu2-buffer.\_sum)\*n2/n;  
+ buffer=(KahanObject) aggOp.increOp.fn.execute(buffer, toadd);  
+ setValue(r, c, buffer.\_sum);  
+ setValue(r+1, c, n);  
+ setValue(r+2, c, buffer.\_correction);  
+ }  
+ }else if(aggOp.correctionLocation==4)  
+ {  
+ double n, n2, mu2;  
+ for(int r=0; r<rlen; r++)  
+ for(int c=0; c<clen-2; c++)  
+ {  
+ buffer.\_sum=this.getValue(r, c);  
+ n=this.getValue(r, c+1);  
+ buffer.\_correction=this.getValue(r, c+2);  
+ mu2=newWithCor.getValue(r, c);  
+ n2=newWithCor.getValue(r, c+1);  
+ n=n+n2;  
+ double toadd=(mu2-buffer.\_sum)\*n2/n;  
+ buffer=(KahanObject) aggOp.increOp.fn.execute(buffer, toadd);  
+ setValue(r, c, buffer.\_sum);  
+ setValue(r, c+1, n);  
+ setValue(r, c+2, buffer.\_correction);  
+ }  
+ }  
+ else  
+ throw new DMLRuntimeException("unrecognized correctionLocation: "+aggOp.correctionLocation);  
+ }  
========================================

TAMAYA-3: Fixed failing tests, fixed infinite loop.  
Updated Design-documentation.  
  
Project: incubator-tamaya.txt  
Commit: 6a489cb60c51a40209048d9cb46bd0723b477c2f  
ClassName: org.apache.tamaya.ucs.UC1ReadProperties.java  
  
+ public void checkUndefined() {  
+ PropertyProvider provider = PropertyProviders.fromPaths("classpath:ucs/UC1ReadProperties/UC1ReadPropertiesTest.properties");  
**+ assertFalse(provider.get("blabla").isPresent());  
+ assertTrue(provider.get("a.b.c").isPresent());**+ }  
========================================

TAMAYA-3: Fixed failing tests, fixed infinite loop.  
Updated Design-documentation.  
  
Project: incubator-tamaya.txt  
Commit: 6a489cb60c51a40209048d9cb46bd0723b477c2f  
ClassName: org.apache.tamaya.ucs.UC1ReadProperties.java  
  
+ public void checkPassDefaultValues() {  
+ PropertyProvider provider = PropertyProviders.fromPaths("classpath:ucs/UC1ReadProperties/UC1ReadPropertiesTest.properties");  
**+ assertEquals("myDefaultValue", provider.get("blabla").orElse("myDefaultValue"));**+ }  
========================================

TAMAYA-3: Fixed failing tests, fixed infinite loop.  
Updated Design-documentation.  
  
Project: incubator-tamaya.txt  
Commit: 6a489cb60c51a40209048d9cb46bd0723b477c2f  
ClassName: org.apache.tamaya.ucs.UC1ReadProperties.java  
  
+ public void checkMetaInfo() {  
+ PropertyProvider provider = PropertyProviders.fromPaths("classpath:ucs/UC1ReadProperties/UC1ReadPropertiesTest.properties");  
+ MetaInfo meta = provider.getMetaInfo();  
**+ assertNotNull(meta);**+ }  
========================================

TAMAYA-3: Fixed failing tests, fixed infinite loop.  
Updated Design-documentation.  
  
Project: incubator-tamaya.txt  
Commit: 6a489cb60c51a40209048d9cb46bd0723b477c2f  
ClassName: org.apache.tamaya.ucs.UC1ReadProperties.java  
  
+ public void checkNeverNull() {  
+ PropertyProvider provider = PropertyProviders.fromPaths("classpath:ucs/UC1ReadProperties/UC1ReadPropertiesTest.properties");  
**+ assertNotNull(provider.get("blabla"));  
+ assertNotNull(provider.get("a.b.c"));**+ }  
========================================

TAMAYA-3: Fixed failing tests, fixed infinite loop.  
Updated Design-documentation.  
  
Project: incubator-tamaya.txt  
Commit: 6a489cb60c51a40209048d9cb46bd0723b477c2f  
ClassName: org.apache.tamaya.ucs.UC1ReadProperties.java  
  
+ public void checkToMap() {  
+ PropertyProvider provider = PropertyProviders.fromPaths("classpath:ucs/UC1ReadProperties/UC1ReadPropertiesTest.properties");  
+ Map<String, String> map = provider.toMap();  
**+ assertNotNull(map);  
+ assertEquals(map.size(), 16);  
+ assertEquals(provider.keySet(), map.keySet());  
+ assertTrue(map.keySet().contains("boolean"));  
+ assertFalse(map.keySet().contains("blabla"));**+ }  
========================================

TAMAYA-3: Fixed failing tests, fixed infinite loop.  
Updated Design-documentation.  
  
Project: incubator-tamaya.txt  
Commit: 6a489cb60c51a40209048d9cb46bd0723b477c2f  
ClassName: org.apache.tamaya.ucs.UC1ReadProperties.java  
  
+ public void checkForAValue() {  
+ PropertyProvider provider = PropertyProviders.fromPaths("classpath:ucs/UC1ReadProperties/UC1ReadPropertiesTest.properties");  
**+ assertFalse(provider.containsKey("blabla"));  
+ assertTrue(provider.containsKey("num.BD"));  
+ assertFalse(provider.get("blabla").isPresent());  
+ assertTrue(provider.get("num.BD").isPresent());**+ }  
========================================

TAMAYA-3: Fixed failing tests, fixed infinite loop.  
Updated Design-documentation.  
  
Project: incubator-tamaya.txt  
Commit: 6a489cb60c51a40209048d9cb46bd0723b477c2f  
ClassName: org.apache.tamaya.ucs.UC1ReadProperties.java  
  
+ public void checkKeys() {  
+ PropertyProvider provider = PropertyProviders.fromPaths("classpath:ucs/UC1ReadProperties/UC1ReadPropertiesTest.properties");  
**+ assertEquals(provider.keySet().size(), 16);  
+ assertTrue(provider.keySet().contains("boolean"));  
+ assertFalse(provider.keySet().contains("blabla"));**+ }  
========================================

fixed bugs in input event processing  
  
Project: incubator-taverna-engine.txt  
Commit: 8e4e2ea23b522a37ee7619c1214cfd092ef02095  
ClassName: net.sf.taverna.t2.provenance.lineageservice.ProvenanceAnalysis.java  
  
+ private void xformStep(String wfID,   
+ Var outputVar, // we need the dnl from this output var  
+ String proc,  
+ String path,  
+ Set<String> selectedProcessors,   
+ List<LineageSQLQuery> lqList   
+ ) throws SQLException {  
+ Map<String, String> varsQueryConstraints = new HashMap<String, String>();  
+ List<Var> inputVars = null;  
+ if (getPq().isDataflow(proc)) { // if we are looking at the output of an entire dataflow  
+ inputVars = new ArrayList<Var>();  
+ inputVars.add(outputVar);  
+ } else if (proc.equals(OUTPUT\_CONTAINER\_PROCESSOR)) { // same action as prev case, but may change in the future  
+ inputVars = new ArrayList<Var>();  
+ inputVars.add(outputVar);  
+ } else {  
+ varsQueryConstraints.put("W.instanceID", wfID);  
+ varsQueryConstraints.put("pnameRef", proc);   
+ varsQueryConstraints.put("inputOrOutput", "1");   
+ inputVars = getPq().getVars(varsQueryConstraints);  
+ }  
+ Map<Var,String> var2Path = new HashMap<Var,String>();  
+ Map<Var,Integer> var2delta = new HashMap<Var,Integer>();  
+ if (path == null) { // nothing to split  
+ for (Var inputVar: inputVars) var2Path.put(inputVar, null);  
+ } else {  
+ int minPathLength = 0; // if input path is shorter than this we give up granularity altogether  
+ for (Var inputVar: inputVars) {  
+ int delta = inputVar.getActualNestingLevel() - inputVar.getTypeNestingLevel();  
+ var2delta.put(inputVar, new Integer(delta));  
+ minPathLength += delta;  
+ }  
+ String iterationVector[] = path.split(",");  
+ if (iterationVector.length < minPathLength) { // no path is propagated  
+ for (Var inputVar: inputVars) {  
+ var2Path.put(inputVar, null);  
+ }  
+ } else { // compute projected paths  
+ String[] projectedPath;   
+ int start = 0;  
+ for (Var inputVar: inputVars) {  
+ int projectedPathLength = var2delta.get(inputVar); // this is delta   
+ if (projectedPathLength > 0) { // this var is involved in iteration  
+ projectedPath = new String[projectedPathLength];  
+ for (int i=0; i<projectedPathLength; i++) {   
+ projectedPath[i] = iterationVector[start+i];  
+ }   
+ start += projectedPathLength;  
+ StringBuffer iterationFragment = new StringBuffer();  
+ for (String s:projectedPath) { iterationFragment.append(s+","); }  
+ iterationFragment.deleteCharAt(iterationFragment.length()-1);  
+ var2Path.put(inputVar, iterationFragment.toString());  
+ } else { // associate empty path to this var  
+ var2Path.put(inputVar, null);  
+ }  
+ }  
+ }  
+ }  
+ currentPath.add(proc);  
+ if (selectedProcessors.contains(proc)) {   
+ List<List<String>> paths = validPaths.get(proc);  
+ List<String> pathCopy = new ArrayList<String>();  
+ for (String s:currentPath) {  
+ if (!getPq().isDataflow(s)) pathCopy.add(s);  
+ }   
+ paths.add(pathCopy);   
+ }  
+ Map<String, ProvenanceArtifact> var2Artifact = new HashMap<String, ProvenanceArtifact>();  
+ Map<String, ProvenanceRole> var2ArtifactRole = new HashMap<String, ProvenanceRole>();  
+ if (selectedProcessors.isEmpty() || selectedProcessors.contains(proc)) {  
+ List<LineageSQLQuery> newLqList = getPq().lineageQueryGen(wfID, proc, var2Path, outputVar, path, isReturnOutputs() || var2Path.isEmpty());  
+ lqList.addAll(newLqList);  
+ boolean doOPM = (aOPMManager != null); // any problem below will set this to false  
+ String role = null;  
+ VarBinding vb = null;  
+ String URIFriendlyIterationVector =null;  
+ if (doOPM) {  
+ Map<String, String> vbConstraints = new HashMap<String, String>();  
+ vbConstraints.put("VB.PNameRef", outputVar.getPName());  
+ vbConstraints.put("VB.varNameRef", outputVar.getVName());  
+ vbConstraints.put("VB.wfInstanceRef", wfID);  
+ if (path != null) {   
+ if (path.startsWith("["))   
+ vbConstraints.put("VB.iteration", path);  
+ else  
+ vbConstraints.put("VB.iteration", "["+path+"]");  
+ }  
+ List<VarBinding> vbList = getPq().getVarBindings(vbConstraints); // DB  
+ if (vbList == null || vbList.size()==0) {  
+ logger.debug("no entry corresponding to conditions: proc="+  
+ outputVar.getPName()+" var = "+outputVar.getVName()+" iteration = "+path);  
+ doOPM = false;  
+ } else {  
+ vb = vbList.get(0);  
+ URIFriendlyIterationVector = vb.getIteration().  
+ replace(',', '-').replace('[', ' ').replace(']', ' ').trim();  
+ if (URIFriendlyIterationVector.length()>0) {  
+ role = vb.getPNameRef()+"/"+vb.getVarNameRef()+"?it="+URIFriendlyIterationVector;  
+ } else  
+ role = vb.getPNameRef()+"/"+vb.getVarNameRef();  
+ if (aOPMManager!=null && !pq.isDataflow(proc)) {  
+ if (isRecordArtifactValues())  
+ aOPMManager.addArtifact(vb.getValue(), vb.getResolvedValue());  
+ else   
+ aOPMManager.addArtifact(vb.getValue());  
+ aOPMManager.createRole(role);  
+ }  
+ aOPMManager.addProcess(proc, vb.getIteration(), URIFriendlyIterationVector);  
**+ aOPMManager.assertGeneratedBy(**+ aOPMManager.getCurrentArtifact(),   
+ aOPMManager.getCurrentProcess(),   
+ aOPMManager.getCurrentRole(),   
+ aOPMManager.getCurrentAccount(),  
+ true); // true -> prevent duplicates CHECK  
+ }  
+ }  
+ for (LineageSQLQuery lq: newLqList) {  
+ LineageQueryResult inputs = getPq().runLineageQuery(lq, isIncludeDataValue());  
  
+ for (LineageQueryResultRecord resultRecord: inputs.getRecords()) {  
+ if (!resultRecord.isInput()) continue;  
+ URIFriendlyIterationVector = resultRecord.getIteration().  
+ replace(',', '-').replace('[', ' ').replace(']', ' ').trim();  
+ boolean found = false; // used to avoid duplicate process resources  
+ if (isRecordArtifactValues())   
+ aOPMManager.addArtifact(resultRecord.getValue(), resultRecord.getResolvedValue());  
+ else   
+ aOPMManager.addArtifact(resultRecord.getValue());  
+ var2Artifact.put(resultRecord.getVname(), aOPMManager.getCurrentArtifact());  
+ if (URIFriendlyIterationVector.length()>0) {  
+ role = resultRecord.getPname()+"/"+resultRecord.getVname()+"?it="+URIFriendlyIterationVector;  
+ } else  
+ role = resultRecord.getPname()+"/"+resultRecord.getVname();  
+ aOPMManager.createRole(role); // this also sets currentRole to role   
+ var2ArtifactRole.put(resultRecord.getVname(), aOPMManager.getCurrentRole());  
**+ aOPMManager.assertUsed(**+ aOPMManager.getCurrentArtifact(),   
+ aOPMManager.getCurrentProcess(),   
+ aOPMManager.getCurrentRole(),   
+ aOPMManager.getCurrentAccount(),  
+ true); // true -> prevent duplicates CHECK   
+ }  
+ }  
+ }  
+ }  
+ for (Var inputVar: inputVars) {  
+ xferStep(wfID, inputVar.getVName(), inputVar.getPName(), var2Path.get(inputVar), selectedProcessors, lqList);   
+ }  
+ currentPath.remove(currentPath.size()-1); // CHECK   
+ } // end xformStep  
========================================

fixed minor testing problems. fixed some OPM details. Added DataValueExtractor pattern  
  
Project: incubator-taverna-engine.txt  
Commit: c693bd73321b915b65ba0c7bd968ad0b7b97afe5  
ClassName: net.sf.taverna.t2.provenance.lineageservice.utils.OPMManager.java  
  
@@ -80,2 +131,15 @@ public void addArtifact(String aName) {  
+ public void addArtifact(String aName) {  
+ Resource r = Resource.uriRef(aName);  
+ currentArtifact = graph.newArtifact(aName, r);  
**+ graph.assertArtifact(currentArtifact);**+ }  
========================================

ZEPPELIN-148 Notebook scope AngularObject for Angular display system  
Described https://issues.apache.org/jira/browse/ZEPPELIN-148  
Author: Lee moon soo <moon@apache.org>  
Closes #143 from Leemoonsoo/localscope and squashes the following commits:  
51340d2 [Lee moon soo] Increase paragraph run timeout  
e00a9af [Lee moon soo] Fix test  
80ff3d4 [Lee moon soo] Fix style  
ead4f70 [Lee moon soo] Front end interaction test with angularBind, angularUnbind, angularWatch  
c7c63d1 [Lee moon soo] make sure test is executed  
d74820f [Lee moon soo] Enabling selenium test  
b817bd9 [Lee moon soo] Cleanup the message  
bcb60f8 [Lee moon soo] Clear object correctly  
12bb15a [Lee moon soo] Fix style  
699defd [Lee moon soo] Handle ANGULAR\_OBJECT\_REMOVE event in front-end.  
1c695c0 [Lee moon soo] Send angularObject add/remove event from ZeppelinServer to remote process  
195a8c9 [Lee moon soo] Add Unbind api for global scope object  
378df68 [Lee moon soo] Remove angularObject when notebook removal  
3548cb5 [Lee moon soo] Fix test  
093870c [Lee moon soo] save / load angular object  
25cf5da [Lee moon soo] Take all objects  
bb757e3 [Lee moon soo] Introduce scope (global/local) for angular object  
  
Project: incubator-zeppelin.txt  
Commit: 201ce8dee7bfcdb4503ec1761cf666670138f14f  
ClassName: org.apache.zeppelin.ZeppelinIT.java  
  
- public void testAnnotationStmt() {  
- WebDriver driver = getWebDriver();  
- try {  
- WebElement start = driver.findElement(By.partialLinkText("Start"));  
- start.click();  
+ driver.findElement(By.xpath(  
+ getParagraphXPath(1) + "//div[@id=\"angularTestButton\"]")).click();  
- (new WebDriverWait(driver, 10)).until(new ExpectedCondition<Boolean>() {  
- public Boolean apply(WebDriver d) {  
- return d.findElement(By.linkText("Create new Job")).isDisplayed();  
- }  
- });  
**+ assertEquals("BindingTest\_3\_", driver.findElement(By.xpath(**+ getParagraphXPath(1) + "//div[@id=\"angularTestButton\"]")).getText());  
+ waitForParagraph(3, "FINISHED");  
- driver.findElement(By.linkText("Create new Job")).click();  
**+ assertEquals("myVar=3", driver.findElement(By.xpath(**+ getParagraphXPath(3) + "//div[@ng-bind=\"paragraph.result.msg\"]")).getText());  
- (new WebDriverWait(driver, 10)).until(new ExpectedCondition<Boolean>() {  
- public Boolean apply(WebDriver d) {  
- return d.findElement(By.linkText("Run")).isDisplayed();  
- }  
- });  
+ WebElement paragraph5Editor = driver.findElement(By.xpath(getParagraphXPath(5) + "//textarea"));  
+ paragraph5Editor.sendKeys(  
+ "z.angularUnbind" + Keys.chord(Keys.SHIFT, "9") + "\"myVar\")");  
+ paragraph5Editor.sendKeys(Keys.chord(Keys.SHIFT, Keys.ENTER));  
+ waitForParagraph(5, "FINISHED");  
- driver.findElement(By.xpath("//div[@id='zqlEditor']//textarea")).sendKeys("@driver set exec;");  
- driver.findElement(By.xpath("//div[@id='zqlEditor']//textarea")).sendKeys("\necho 'hello world';");  
**+ assertEquals("BindingTest\_\_", driver.findElement(By.xpath(**+ getParagraphXPath(1) + "//div[@id=\"angularTestButton\"]")).getText());  
- driver.findElement(By.xpath("//div[@id='zqlEditor']//textarea")).sendKeys(Keys.chord(Keys.COMMAND, Keys.ENTER));  
- driver.findElement(By.xpath("//div[@id='zqlEditor']//textarea")).sendKeys(Keys.chord(Keys.CONTROL, Keys.ENTER));  
- driver.findElement(By.linkText("Run")).click();  
+ paragraph2Editor = driver.findElement(By.xpath(getParagraphXPath(2) + "//textarea"));  
+ paragraph2Editor.sendKeys(Keys.chord(Keys.SHIFT, Keys.ENTER));  
+ waitForParagraph(2, "FINISHED");  
- (new WebDriverWait(driver, 10)).until(new ExpectedCondition<Boolean>() {  
- public Boolean apply(WebDriver d) {  
- return d.findElement(By.xpath("//div//a[text()='Running ...']")).isDisplayed();  
- }  
- });  
**+ assertEquals("BindingTest\_1\_", driver.findElement(By.xpath(**+ getParagraphXPath(1) + "//div[@id=\"angularTestButton\"]")).getText());  
- (new WebDriverWait(driver, 60)).until(new ExpectedCondition<Boolean>() {  
- public Boolean apply(WebDriver d) {  
- return d.findElement(By.xpath("//div//a[text()='Run']")).isDisplayed();  
+ System.out.println("testCreateNotebook Test executed");  
 }  
- });  
- WebElement msg = driver.findElement(By.id("msgBox"));  
- if (msg!=null) {  
- System.out.println("msgBox="+msg.getText());  
+ private String createNewNoteAndGetName() {  
+ List<WebElement> notebookLinks = driver.findElements(By  
+ .xpath("//div[contains(@class, \"col-md-4\")]/div/ul/li"));   
+ List<String> notebookTitles = new LinkedList<String>();  
+ for (WebElement el : notebookLinks) {  
+ notebookTitles.add(el.getText());  
 }  
- (new WebDriverWait(driver, 20)).until(new ExpectedCondition<Boolean>() {  
- public Boolean apply(WebDriver d) {  
- return d.findElement(By.xpath("//div[@id='visualizationContainer']//iframe")).isDisplayed();  
+ WebElement createNoteLink = driver.findElement(By.partialLinkText("Create new note"));  
+ createNoteLink.click();  
+ try {  
+ Thread.sleep(500); // wait for notebook list updated  
+ } catch (InterruptedException e) {  
 }   
- });  
- WebDriver iframe = driver.switchTo().frame(driver.findElement(By.xpath("//div[@id='visualizationContainer']//iframe")));  
+ List<WebElement> notebookLinksAfterCreate = driver.findElements(By  
+ .xpath("//div[contains(@class, \"col-md-4\")]/div/ul/li"));  
- (new WebDriverWait(iframe, 20)).until(new ExpectedCondition<Boolean>() {  
- public Boolean apply(WebDriver d) {  
- return d.findElement(By.xpath("//table//td[text()='hello world']")).isDisplayed();  
+ Iterator<WebElement> it = notebookLinksAfterCreate.iterator();  
+ while (it.hasNext()) {  
+ WebElement newEl = it.next();  
+ if (notebookTitles.contains(newEl.getText())) {  
+ it.remove();  
 }  
- });  
- } catch (WebDriverException e){  
- File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);  
- System.out.println("Screenshot in: " + scrFile.getAbsolutePath());  
- throw e;  
- } finally {  
- driver.quit();  
 }  
**+ assertEquals(1, notebookLinksAfterCreate.size());**+ return notebookLinksAfterCreate.get(0).getText();  
 }  
 }  
========================================

ZEPPELIN-61 Lens Interpreter  
Add Lens interpreter. #100 has implementation from praagarw.  
This pr grabbing a necessary commits from #100 and rebase from current master branch.  
Author: Pranav Agarwal <praagarw@gmail.com>  
Author: Lee moon soo <moon@apache.org>  
Closes #114 from Leemoonsoo/ZEPPELIN-61 and squashes the following commits:  
0edaf8c [Lee moon soo] Bumpup lens interpreter artifact version  
adc23f9 [Pranav Agarwal] fixed merge conflicts for conf/zeppelin-site.xml.template  
97eb569 [Pranav Agarwal] added lens entry to conf/zeppelin-site.xml.template  
b81ab78 [Pranav Agarwal] Lens Interpreter  
  
Project: incubator-zeppelin.txt  
Commit: 1bc5e8df922aff6332c9b5365f71944ca063fd01  
ClassName: org.apache.zeppelin.lens.LensSimpleExecutionStrategy.java  
  
+ public Object execute(ParseResult parseResult) throws RuntimeException {  
**+ Assert.notNull(parseResult, "Parse result required");**+ logger.info("LensSimpleExecutionStrategy execute method invoked");  
+ synchronized (this) {  
**+ Assert.isTrue(isReadyForCommands(), "SimpleExecutionStrategy not yet ready for commands");**+ Object target = parseResult.getInstance();  
+ if (target instanceof ExecutionProcessor) {  
+ ExecutionProcessor processor = ((ExecutionProcessor) target);  
+ parseResult = processor.beforeInvocation(parseResult);  
+ try {  
+ Object result = invoke(parseResult);  
+ processor.afterReturningInvocation(parseResult, result);  
+ return result;  
+ } catch (Throwable th) {  
+ processor.afterThrowingInvocation(parseResult, th);  
+ return handleThrowable(th);  
+ }  
+ }  
+ else {  
+ return invoke(parseResult);  
+ }  
+ }  
+ }  
========================================

ZEPPELIN-41 fix: zengine module clenup: all warnings fixed  
  
Project: incubator-zeppelin.txt  
Commit: 51391e5cc389e127aa0d7a78a03491296ede394f  
ClassName: com.nflabs.zeppelin.zengine.L.java  
  
@@ -266,3 +251,3 @@ public Z pipe(Z z){  
 } else {  
**+ assert false; // should never reach here** }  
========================================

ZEPPELIN-25 Ability to create rich gui inside of Notebook  
This PR implements https://issues.apache.org/jira/browse/ZEPPELIN-25  
Here's a short video demo of this feature.  
[![IMAGE ALT TEXT HERE](http://img.youtube.com/vi/xU5TBS\_MsAs/0.jpg)](http://www.youtube.com/watch?v=xU5TBS\_MsAs)  
for someone who want to try, here's api  
```scala  
// bind 'varName' variable with 'v' value  
z.angularBind(varName: String, v: Object)  
// unbind 'varName'  
z.angularUnbind(varName: String)  
// get value of 'varName'  
z.angular(varName:String)  
// add watcher to 'varName' variable.  
// that is monitoring change and run 'func' when it is changed.  
z.angularWatch(varName:String, func: (Object, Object) => Unit))  
// remove watcher from 'varName'  
z.angularUnwatch(varName:String)  
```  
Any paragraph's output starting with '%angular' will be considered as angular code. %angular as a interpreter also available.  
![image](https://cloud.githubusercontent.com/assets/1540981/7003457/a1e6fe2a-dc95-11e4-8272-380f11c6ae81.png)  
Any feedback is welcome!  
Author: Lee moon soo <moon@apache.org>  
Closes #27 from Leemoonsoo/angular and squashes the following commits:  
04d7175 [Lee moon soo] Remove implicit conversion because of side effect  
34fa298 [Lee moon soo] jquery to angular  
8076098 [Lee moon soo] Remove unnecessary type information  
88fd635 [Lee moon soo] catch and print watcher user provided routine exception  
46dba2f [Lee moon soo] Catch sql syntax error  
2ebfa59 [Lee moon soo] Let z.run optionally take InterpreterContext  
ee29866 [Lee moon soo] ZEPPELIN-32 implement z.show()  
dac416d [Lee moon soo] Implement z.run()  
0899011 [Lee moon soo] Fix test  
0033d32 [Lee moon soo] Add angular interpreter  
42ee479 [Lee moon soo] com.nflabs -> org.apache  
4d32d19 [Lee moon soo] ZEPPELIN-25 prevent watcher called multiple times  
d4d270e [Lee moon soo] ZEPPELIN-25 add unittest  
6ce8f36 [Lee moon soo] ZEPPELIN-25 implement watcher  
6df7f23 [Lee moon soo] ZEPPELIN-25 broadcast angular object change to related notes  
5954e29 [Lee moon soo] ZEPPELIN-25 save/restore angular object registry snapshot to the notebook file  
c288198 [Lee moon soo] ZEPPELIN-25 send scope variables when loading note  
67f6926 [Lee moon soo] ZEPPELIN-25 impelemnet JS(angular) -JVM(scala) two-way binding  
bb52d7b [Lee moon soo] Add %angular display system  
a7c77b8 [Lee moon soo] Update license of ScreenCaptureHtmlUnitDriver.java  
6d7e063 [Lee moon soo] Add source file license header  
  
Project: incubator-zeppelin.txt  
Commit: 58b70e3bc0b9fb814a5919037318a79ae67f678f  
ClassName: org.apache.zeppelin.interpreter.remote.RemoteInterpreterContextRunner.java  
  
+ public void run() {  
**+ throw new InterpreterException("Assert");**+ }  
========================================

Completed reorganisation of packages and fixed most of the tests (not the two object store tests.  
  
Project: isis.txt  
Commit: 94996921b06afb56f8ddd3aea00c0274d28ed5df  
ClassName: org.nakedobjects.object.persistence.objectstore.ObjectStorePersitenceManager.java  
  
+ public void reload(NakedObject object) {}  
+ public void resolveImmediately(NakedObject object) {  
+ ResolveState resolveState = object.getResolveState();  
+ if (resolveState.isResolvable(ResolveState.RESOLVING)) {  
**+ Assert.assertFalse("only resolve object that is not yet resolved", object, object.getResolveState().isResolved());  
+ Assert.assertTrue("only resolve object that is persistent", object, object.getResolveState().isPersistent());**+ LOG.info("resolve-immediately: " + object);  
+ objectStore.resolveImmediately(object);  
+ }  
+ }  
========================================

ISIS-14: changes to runtime for JDO  
\* generalized support to handle javax.jdo.identity.\*Identity objectIds.  
\* made OidMarshaller be accessed from IsisContext  
\* extended Oid so that can optionally store version (a long); this functionality is not yet used anywhere  
\* fixed bug with wicket with respect to finder actions.  
  
Project: isis.txt  
Commit: 41a9db1551151f71aa62a02d6a1f9a3fa0b89ec5  
ClassName: org.apache.isis.core.metamodel.adapter.oid.OidMarshaller\_roundtripping.java  
  
+ public void rootOid\_withNoVersion() {  
+ RootOid oid = RootOidDefault.create(ObjectSpecId.of("CUS"), "123");  
+ final String enString = oid.enString(oidMarshaller);  
+ final RootOid deString = RootOidDefault.deString(enString, oidMarshaller);  
**+ assertThat(deString, is(oid));**+ }  
========================================

Completed reorganisation of packages and fixed most of the tests (not the two object store tests.  
  
Project: isis.txt  
Commit: 94996921b06afb56f8ddd3aea00c0274d28ed5df  
ClassName: org.nakedobjects.object.persistence.objectstore.ObjectStorePersitenceManager.java  
  
+ public void addObjectChangedListener(DirtyObjectSet listener) {  
**+ Assert.assertNotNull("must set a listener", listener);**+ this.objectsToRefreshViewsFor = listener;  
+ }  
========================================

ISIS-463: can now intermix @unit and @integration level specs.  
\* renamed ScenarioExecution's get and put to getVar and putVar  
\* added a removeVar  
\* updated corresponding convenience methods in IntegationTestAbstract and CukeStepDefsAbstract  
\* added supportsMocks into CukeStepDefsAbstract  
\* refactored todo specs with guards (necessary, unfortunately)  
- InMemoryDB utility class refactorings  
- fixed license header comments.  
  
Project: isis.txt  
Commit: 3d271d24ca5fedbdbf796fd069a061920ba2e7fa  
ClassName: integration.specs.todoitem.ToDoItemStepDefs.java  
  
+ public void I\_mark\_it\_as\_not\_yet\_complete() throws Throwable {  
+ ToDoItem toDoItem = getVar(null, "toDoItem", ToDoItem.class);  
**+ assertThat(toDoItem.isComplete(), is(true));**+ toDoItem.setComplete(false);  
+ }  
========================================

Fixed issues with the distribution and tidied it up some.  
  
Project: isis.txt  
Commit: 2f26c062ed64a87a1b74f13e40d7fb718fb07c19  
ClassName: org.nakedobjects.distribution.ObjectEncoder.java  
  
- private final ObjectData createObjectData(  
+ private final ReferenceData serializeObject(NakedObject object, boolean recursePersistentObjects, int depth) {  
**+ Assert.assertNotNull(object);**+ return (ReferenceData) serializeObject2(object, recursePersistentObjects, depth, new KnownTransients());  
+ }  
========================================

Completed reorganisation of packages and fixed most of the tests (not the two object store tests.  
  
Project: isis.txt  
Commit: 3f2b010d127296b26b96881bb397b47d719a0594  
ClassName: org.nakedobjects.object.loader.ObjectLoaderImpl.java  
  
+ public NakedObject recreateAdapterForPersistent(Oid oid, NakedObjectSpecification specification) {  
**+ Assert.assertNotNull("must have an OID", oid);  
+ Assert.assertTrue("must be an object", specification.isObject());**+ if (isIdentityKnown(oid)) {  
+ return getAdapterFor(oid);  
+ }  
+ LOG.debug("recreating object " + specification.getFullName() + "/" + oid);  
+ Object object = objectFactory.createObject(specification);  
**+ Assert.assertNotNull(oid);  
+ Assert.assertFalse("Identity Map already contains object for OID " + oid, identityAdapterMap.containsKey(oid));**+ PojoAdapter adapter = (PojoAdapter) createObjectAdapter(object);  
+ addIdentityMapping(oid, adapter);  
**+ Assert.assertTrue(pojoAdapterMap.getPojo(object) == adapter);  
+ Assert.assertTrue(identityAdapterMap.get(oid) == adapter);**+ adapter.recreatedAs(oid);  
+ return adapter;  
+ }  
========================================

Completed reorganisation of packages and fixed most of the tests (not the two object store tests.  
  
Project: isis.txt  
Commit: 3f2b010d127296b26b96881bb397b47d719a0594  
ClassName: org.nakedobjects.object.loader.ObjectLoaderImpl.java  
  
+ public NakedObject getAdapterFor(final Object object) {  
**+ Assert.assertNotNull("can't get an adapter for null", this, object);**+ NakedObject adapter = (NakedObject) pojoAdapterMap.getPojo(object);  
+ return adapter;  
+ }  
========================================

Refactored, and added, tests to check resolving issues.  
  
Project: isis.txt  
Commit: 5a8d10da878f44fb5f6eaaa183cb8a61d2b6da1c  
ClassName: org.nakedobjects.utility.ExpectedCalls.java  
  
+ public void addActualMethod(String name) {  
**+ assertLastMethodsParametersCorrect();**+ ExpectedCall actual = new ExpectedCall(name);  
+ actualObjects.addElement(actual);  
**+ assertExpectedNoMoreThanActuals();**+ int element = actualObjects.size() - 1;  
+ ExpectedCall expected = (ExpectedCall) expectedObjects.elementAt(element);  
**+ Assert.assertEquals("Actual method does not match expected.\n",**+ expected.name, name);  
+ }  
========================================

ISIS-233: more tck tests for RO viewer.  
\* primitive/wrapper properties domain object  
\* utility methods  
\* list repr, elements  
\* fix for TypedOfFacet for actions that return List<X>  
  
Project: isis.txt  
Commit: 430e8b3cd174d77fd0406ebbbe574f3090986510  
ClassName: org.apache.isis.viewer.restfulobjects.tck.domainobject.oid.Get\_givenDisabledMembers\_thenRepresentation\_ok\_TODO.java  
  
+ private DomainObjectRepresentation givenDomainObjectRepresentationFor(final String domainType, String instanceId) throws JsonParseException, JsonMappingException, IOException {  
+ final DomainObjectResource domainObjectResource = client.getDomainObjectResource();  
+ final Response domainObjectResp = domainObjectResource.object(domainType, instanceId);  
+ final RestfulResponse<DomainObjectRepresentation> domainObjectJsonResp = RestfulResponse.ofT(domainObjectResp);  
**+ assertThat(domainObjectJsonResp.getStatus().getFamily(), is(Family.SUCCESSFUL));**+ final DomainObjectRepresentation domainObjectRepr = domainObjectJsonResp.getEntity();  
+ return domainObjectRepr;  
+ }  
========================================

ISIS-14: changes to runtime for JDO  
\* generalized support to handle javax.jdo.identity.\*Identity objectIds.  
\* made OidMarshaller be accessed from IsisContext  
\* extended Oid so that can optionally store version (a long); this functionality is not yet used anywhere  
\* fixed bug with wicket with respect to finder actions.  
  
Project: isis.txt  
Commit: 41a9db1551151f71aa62a02d6a1f9a3fa0b89ec5  
ClassName: org.apache.isis.core.metamodel.adapter.oid.OidMarshaller\_roundtripping.java  
  
+ public void rootOid\_withVersion() {  
+ RootOid oid = RootOidDefault.create(ObjectSpecId.of("CUS"), "123", 90807L);  
+ final String enString = oid.enString(oidMarshaller);  
+ final RootOid deString = RootOidDefault.deString(enString, oidMarshaller);  
**+ assertThat(deString, is(oid));  
+ assertThat(deString.getVersion(), is(oid.getVersion())); // assert separately because not part of equality check**+ }  
========================================

work in progress  
- version always protected  
- versionHistory always protected  
- fix Node.getReferences && reference properties  
- mixinTypes are 'active' upon save only (and not before). therefore  
cleanup after removeMixin is done only after save.  
adding new states defined by a new mixin are only added after the  
node has been saved.  
- nodestates identified by a uuid only were created a second time upon  
ChildNodeEntry.resolve => fix needs to be improved  
- Events/ChangeLog are processed by the save-target state  
- ChangeLog.persisted not used any more  
  
Project: jackrabbit.txt  
Commit: 9ae8bbb5c63832a2db95f20a916c3c350f4f0681  
ClassName: org.apache.jackrabbit.jcr2spi.state.NodeState.java  
  
**+ private void assertAvailability() {**+ if (uuid != null) {  
+ ChildNodeEntry cne = parent.childNodeEntries.get(uuid);  
+ if (!cne.isAvailable()) {  
+ try {  
+ cne.getNodeState();  
+ } catch (ItemStateException e) {  
+ }  
+ }  
+ }  
+ }  
========================================

work in progress  
- version always protected  
- versionHistory always protected  
- fix Node.getReferences && reference properties  
- mixinTypes are 'active' upon save only (and not before). therefore  
cleanup after removeMixin is done only after save.  
adding new states defined by a new mixin are only added after the  
node has been saved.  
- nodestates identified by a uuid only were created a second time upon  
ChildNodeEntry.resolve => fix needs to be improved  
- Events/ChangeLog are processed by the save-target state  
- ChangeLog.persisted not used any more  
  
Project: jackrabbit.txt  
Commit: 9ae8bbb5c63832a2db95f20a916c3c350f4f0681  
ClassName: org.apache.jackrabbit.jcr2spi.state.NodeState.java  
  
+ private static QName[] getMixinNames(PropertyState ps) throws RepositoryException {  
**+ assert QName.JCR\_MIXINTYPES.equals(ps.getQName());**+ QValue[] values = ps.getValues();  
+ QName[] newMixins = new QName[values.length];  
+ for (int i = 0; i < values.length; i++) {  
+ newMixins[i] = QName.valueOf(values[i].getString());  
+ }  
========================================

work in progress  
JCR2SPI  
--------------------------------------------------------------  
- reorder ItemState/NodeState/PropertyState in order to group  
methods by state types (overlayed workspace state vs. session/transient states)  
- add utility methods that allow to determine and validate the  
type of ItemState.  
- move childitementry classes to separate package and force usage  
of interface methods  
- add checks for state type to ItemManager implementation  
- add checks for state type to LockManager  
- fix Node.getMixinTypes()  
- rename ItemState.pull to 'reset'. remove todo asking for a careful  
merging of wsp-State and introduce a new method 'merge'.  
SPI  
--------------------------------------------------------------------  
- rename login as suggested by julian  
- add method dispose(SessionInfo)  
- rename ItemId.getRelativePath to getPath  
  
Project: jackrabbit.txt  
Commit: 9701ef4452c453b211f426d0a357b4f4f0e2d945  
ClassName: org.apache.jackrabbit.jcr2spi.state.ItemState.java  
  
- public boolean hasOverlayedState() {  
- return overlayedState != null;  
+ public void addListener(ItemStateLifeCycleListener listener) {  
+ synchronized (listeners) {  
**+ assert (!listeners.contains(listener));**+ listeners.add(listener);  
+ }  
 }  
========================================

OAK-781: Clarify / fix effects of MISSING\_NODE as base state of NodeBuilder  
rework MemoryNodeBuilder:  
- promote MutableNodeState to top level package private class  
- separate state tracking and hierarchy tracking between MemoryNodeBuilder and MutableNodeState, respectively  
- rename NodeBuilder methods to match those of NodeState  
  
Project: jackrabbit-oak.txt  
Commit: 4a32b32f956c822c5d55296735369ac491a30147  
ClassName: org.apache.jackrabbit.oak.plugins.memory.MutableNodeState.java  
  
+ public PropertyState getProperty(String name) {  
**+ assert base != null;**+ return withProperties(base, properties).getProperty(name);  
+ }  
========================================

OAK-781: Clarify / fix effects of MISSING\_NODE as base state of NodeBuilder  
rework MemoryNodeBuilder:  
- promote MutableNodeState to top level package private class  
- separate state tracking and hierarchy tracking between MemoryNodeBuilder and MutableNodeState, respectively  
- rename NodeBuilder methods to match those of NodeState  
  
Project: jackrabbit-oak.txt  
Commit: 4a32b32f956c822c5d55296735369ac491a30147  
ClassName: org.apache.jackrabbit.oak.plugins.memory.MutableNodeState.java  
  
+ public boolean exists() {  
**+ assert base != null;**+ return base.exists();  
+ }  
========================================

OAK-781: Clarify / fix effects of MISSING\_NODE as base state of NodeBuilder  
rework MemoryNodeBuilder:  
- promote MutableNodeState to top level package private class  
- separate state tracking and hierarchy tracking between MemoryNodeBuilder and MutableNodeState, respectively  
- rename NodeBuilder methods to match those of NodeState  
  
Project: jackrabbit-oak.txt  
Commit: 4a32b32f956c822c5d55296735369ac491a30147  
ClassName: org.apache.jackrabbit.oak.plugins.memory.MutableNodeState.java  
  
+ public String toString() {  
**+ assert base != null;**+ StringBuilder builder = new StringBuilder();  
+ builder.append(base).append(" + {");  
+ String separator = " ";  
+ for (PropertyState property : properties.values()) {  
+ builder.append(separator);  
+ separator = ", ";  
+ builder.append(property);  
+ }  
+ for (Entry<String, MutableNodeState> entry : nodes.entrySet()) {  
+ builder.append(separator);  
+ separator = ", ";  
+ builder.append(entry.getKey()).append(" : ").append(entry.getValue());  
+ }  
+ builder.append(" }");  
+ return builder.toString();  
+ }  
========================================

OAK-781: Clarify / fix effects of MISSING\_NODE as base state of NodeBuilder  
rework MemoryNodeBuilder:  
- promote MutableNodeState to top level package private class  
- separate state tracking and hierarchy tracking between MemoryNodeBuilder and MutableNodeState, respectively  
- rename NodeBuilder methods to match those of NodeState  
  
Project: jackrabbit-oak.txt  
Commit: 4a32b32f956c822c5d55296735369ac491a30147  
ClassName: org.apache.jackrabbit.oak.plugins.memory.MutableNodeState.java  
  
+ public NodeState snapshot() {  
**+ assert base != null;**+ Map<String, NodeState> nodes = newHashMap();  
+ for (Map.Entry<String, MutableNodeState> entry : this.nodes.entrySet()) {  
+ String name = entry.getKey();  
+ MutableNodeState node = entry.getValue();  
+ NodeState before = base.getChildNode(name);  
+ if (node == null) {  
+ if (before.exists()) {  
+ nodes.put(name, null);  
+ }  
+ } else {  
+ NodeState after = node.snapshot();  
+ if (after != before) {  
+ nodes.put(name, after);  
+ }  
+ }  
+ }  
+ return with(base, newHashMap(this.properties), nodes);  
+ }  
========================================

OAK-781: Clarify / fix effects of MISSING\_NODE as base state of NodeBuilder  
rework MemoryNodeBuilder:  
- promote MutableNodeState to top level package private class  
- separate state tracking and hierarchy tracking between MemoryNodeBuilder and MutableNodeState, respectively  
- rename NodeBuilder methods to match those of NodeState  
  
Project: jackrabbit-oak.txt  
Commit: 4a32b32f956c822c5d55296735369ac491a30147  
ClassName: org.apache.jackrabbit.oak.plugins.memory.MutableNodeState.java  
  
+ public boolean hasProperty(String name) {  
**+ assert base != null;**+ return withProperties(base, properties).hasProperty(name);  
+ }  
========================================

OAK-938 UserManager does not honour session refresh settings  
Tentative fix by wrapping the core user manager into a SessionOperation closure  
  
Project: jackrabbit-oak.txt  
Commit: 11d6a44980d476d25a16ce46a0f6fe237ca375fb  
ClassName: org.apache.jackrabbit.oak.jcr.SessionContext.java  
  
 public UserManager getUserManager() {  
 if (userManager == null) {  
- userManager = getConfig(UserConfiguration.class).getUserManager(delegate.getRoot(), namePathMapper);  
+ final UserManager uMgr = getConfig(UserConfiguration.class).getUserManager(delegate.getRoot(), namePathMapper);  
+ userManager = new UserManager() {  
+ @Override  
+ public Authorizable getAuthorizable(final String id) throws RepositoryException {  
+ return delegate.perform(new SessionOperation<Authorizable>() {  
+ @Override  
+ public Authorizable perform() throws RepositoryException {  
+ return uMgr.getAuthorizable(id);  
+ }  
+ });  
+ }  
+ @Override  
+ public Authorizable getAuthorizable(final Principal principal) throws RepositoryException {  
+ return delegate.perform(new SessionOperation<Authorizable>() {  
+ @Override  
+ public Authorizable perform() throws RepositoryException {  
+ return uMgr.getAuthorizable(principal);  
+ }  
+ });  
+ }  
+ @Override  
+ public Authorizable getAuthorizableByPath(final String path) throws UnsupportedRepositoryOperationException, RepositoryException {  
+ return delegate.perform(new SessionOperation<Authorizable>() {  
+ @Override  
+ public Authorizable perform() throws RepositoryException {  
+ return uMgr.getAuthorizableByPath(path);  
+ }  
+ });  
+ }  
+ @Override  
+ public Iterator<Authorizable> findAuthorizables(final String relPath, final String value) throws RepositoryException {  
+ return delegate.perform(new SessionOperation<Iterator<Authorizable>>() {  
+ @Override  
+ public Iterator<Authorizable> perform() throws RepositoryException {  
+ return uMgr.findAuthorizables(relPath, value);  
+ }  
+ });  
+ }  
+ @Override  
+ public Iterator<Authorizable> findAuthorizables(final String relPath, final String value, final int searchType) throws RepositoryException {  
+ return delegate.perform(new SessionOperation<Iterator<Authorizable>>() {  
+ @Override  
+ public Iterator<Authorizable> perform() throws RepositoryException {  
+ return uMgr.findAuthorizables(relPath, value, searchType);  
+ }  
+ });  
+ }  
+ @Override  
+ public Iterator<Authorizable> findAuthorizables(final Query query) throws RepositoryException {  
+ return delegate.perform(new SessionOperation<Iterator<Authorizable>>() {  
+ @Override  
+ public Iterator<Authorizable> perform() throws RepositoryException {  
+ return uMgr.findAuthorizables(query);  
+ }  
+ });  
+ }  
+ @Override  
+ public User createUser(final String userID, final String password) throws AuthorizableExistsException, RepositoryException {  
+ return delegate.perform(new SessionOperation<User>() {  
+ @Override  
+ public User perform() throws RepositoryException {  
+ return uMgr.createUser(userID, password);  
+ }  
+ });  
+ }  
+ @Override  
+ public User createUser(final String userID, final String password, final Principal principal, final String intermediatePath) throws AuthorizableExistsException, RepositoryException {  
+ return delegate.perform(new SessionOperation<User>() {  
+ @Override  
+ public User perform() throws RepositoryException {  
+ return uMgr.createUser(userID, password, principal, intermediatePath);  
+ }  
+ });  
+ }  
+ @Override  
+ public Group createGroup(final String groupID) throws AuthorizableExistsException, RepositoryException {  
+ return delegate.perform(new SessionOperation<Group>() {  
+ @Override  
+ public Group perform() throws RepositoryException {  
+ return uMgr.createGroup(groupID);  
+ }  
+ });  
+ }  
+ @Override  
+ public Group createGroup(final Principal principal) throws AuthorizableExistsException, RepositoryException {  
+ return delegate.perform(new SessionOperation<Group>() {  
+ @Override  
+ public Group perform() throws RepositoryException {  
+ return uMgr.createGroup(principal);  
+ }  
+ });  
+ }  
+ @Override  
+ public Group createGroup(final Principal principal, final String intermediatePath) throws AuthorizableExistsException, RepositoryException {  
+ return delegate.perform(new SessionOperation<Group>() {  
+ @Override  
+ public Group perform() throws RepositoryException {  
+ return uMgr.createGroup(principal, intermediatePath);  
+ }  
+ });  
+ }  
+ @Override  
+ public Group createGroup(final String groupID, final Principal principal, final String intermediatePath) throws AuthorizableExistsException, RepositoryException {  
+ return delegate.perform(new SessionOperation<Group>() {  
+ @Override  
+ public Group perform() throws RepositoryException {  
+ return uMgr.createGroup(groupID, principal, intermediatePath);  
+ }  
+ });  
+ }  
+ @Override  
+ public boolean isAutoSave() {  
+ try {  
+ return delegate.perform(new SessionOperation<Boolean>() {  
+ @Override  
+ public Boolean perform() throws RepositoryException {  
+ return uMgr.isAutoSave();  
+ }  
+ });  
+ } catch (RepositoryException e) {  
**+ assert false : "Unexpected exception: " + e;**+ return false;  
+ }  
+ }  
+ @Override  
+ public void autoSave(final boolean enable) throws UnsupportedRepositoryOperationException, RepositoryException {  
+ delegate.perform(new SessionOperation<Void>() {  
+ @Override  
+ public Void perform() throws RepositoryException {  
+ uMgr.autoSave(enable);  
+ return null;  
+ }  
+ });  
+ }  
+ };  
========================================

OAK-781: Clarify / fix effects of MISSING\_NODE as base state of NodeBuilder  
Explicitly track connectedness through new Head.head()  
  
Project: jackrabbit-oak.txt  
Commit: f6f6507b2e5f42419f489fbd3aec7a4ed4346083  
ClassName: org.apache.jackrabbit.oak.plugins.memory.MemoryNodeBuilder.java  
  
 public MutableNodeState getMutableNodeState() {  
- rootBuilder.head.revision++;  
- return getCurrentNodeState();  
+ MutableNodeState parentState = parent.head().getMutableNodeState();  
+ head = new ConnectedHead(parentState.getMutableChildNode(name));  
+ return head.getMutableNodeState();  
+ }  
+ @Override  
+ public NodeState getImmutableNodeState() {  
**+ assert !(state instanceof MutableNodeState);**========================================

OAK-892: Parameterize oak-jcr tests  
- All tests extending from AbstractRepositoryTest are not parameterized  
- Turn long running ConcurrentAddTest into an integration test  
- Mark test in ConcurrentAddIT as known issue for SegmentMK (OAK-786)  
- Use in memory MicroKernel when possible  
  
Project: jackrabbit-oak.txt  
Commit: d10b95346cc91dc6b01eeb2401d60fd4d5973c3f  
ClassName: org.apache.jackrabbit.oak.jcr.ConcurrentAddIT.java  
  
+ public void addNodes() throws Exception {  
+ assumeTrue(fixture != NodeStoreFixture.SEGMENT\_MK);  
+ List<Exception> exceptions = Collections.synchronizedList(  
+ new ArrayList<Exception>());  
+ Node test = getAdminSession().getRootNode().addNode("test");  
+ List<Thread> worker = new ArrayList<Thread>();  
+ for (int i = 0; i < NUM\_WORKERS; i++) {  
+ String path = test.addNode("node" + i).getPath();  
+ worker.add(new Thread(new Worker(  
+ createAdminSession(), path, exceptions)));  
+ }  
+ getAdminSession().save();  
+ for (Thread t : worker) {  
+ t.start();  
+ }  
+ for (Thread t : worker) {  
+ t.join();  
+ }  
+ for (Exception e : exceptions) {  
+ fail(e.toString());  
+ }  
+ getAdminSession().refresh(false);  
+ for (Node n : new NodeIterable(test.getNodes())) {  
**+ assertEquals(NODES\_PER\_WORKER, Iterables.size(new NodeIterable(n.getNodes())));**+ }  
+ }  
========================================

OAK-781: Clarify / fix effects of MISSING\_NODE as base state of NodeBuilder  
rework MemoryNodeBuilder:  
- promote MutableNodeState to top level package private class  
- separate state tracking and hierarchy tracking between MemoryNodeBuilder and MutableNodeState, respectively  
- rename NodeBuilder methods to match those of NodeState  
  
Project: jackrabbit-oak.txt  
Commit: 4a32b32f956c822c5d55296735369ac491a30147  
ClassName: org.apache.jackrabbit.oak.plugins.memory.MutableNodeState.java  
  
+ public boolean isConnected(String name) {  
**+ assert base != null;**+ return nodes.get(name) != null ||  
+ !nodes.containsKey(name) && base.getChildNode(name).exists();  
+ }  
========================================

OAK-26: MVCC causes write skew  
- test case documenting the issue  
  
Project: jackrabbit-oak.txt  
Commit: 4b7e6b60f57774c9300c8b0e243cf4883e7d710e  
ClassName: org.apache.jackrabbit.oak.jcr.CompatibilityIssues.java  
  
+ public void sessionIsolation() throws RepositoryException {  
+ Repository repository = getRepository();  
+ Session session0 = repository.login();  
+ Node testNode = session0.getNode("/").addNode("testNode");  
+ testNode.setProperty("p1", 1);  
+ testNode.setProperty("p2", 1);  
+ session0.save();  
+ check(getSession());  
+ Session session1 = repository.login();  
+ Session session2 = repository.login();  
+ session1.getNode("/testNode").setProperty("p1", -1);  
+ check(session1);  
+ session1.save();  
+ session2.getNode("/testNode").setProperty("p2", -1);  
+ check(session2); // Throws on JR2, not on JR3  
+ session2.save();  
+ Session session3 = repository.login();  
+ try {  
+ check(session3); // Throws on JR3  
+ fail();  
+ }  
**+ catch (AssertionError e) {**+ }  
+ session0.logout();  
+ session1.logout();  
+ session2.logout();  
+ session3.logout();  
+ }  
========================================

Updates and fixes for existing clients, domain objects and tests, including:  
- Changed MetadataEntry to MetadataValue where required  
**- Added more standardised assertion/error check messages**- Completed CatalogClientLiveTest  
  
Project: jclouds.txt  
Commit: d8390da9b4eeeed871d92fcaf1cd5248be4861df  
ClassName: org.jclouds.vcloud.director.v1\_5.domain.Checks.java  
  
 public static void checkMetadataEntry(MetadataEntry metadataEntry) {+ // Check required fields  
**- assertNotNull(metadataEntry.getKey(), "The Key attribute of a MetadataEntry must be set");  
- assertNotNull(metadataEntry.getValue(), "The Value attribute of a MetadataEntry must be set");  
+ assertNotNull(metadataEntry.getKey(), String.format(NOT\_NULL\_OBJECT\_FMT, "Key", "MetadataEntry"));  
+ assertNotNull(metadataEntry.getValue(), String.format(NOT\_NULL\_OBJECT\_FMT, "Value", "MetadataEntry"));**========================================

issue 384: added DetachDistroMedium and RetrieveBridgedInterface + Test  
  
Project: jclouds.txt  
Commit: 3de92bd2922e719bb8d9e291aad29800c33092c8  
ClassName: org.jclouds.virtualbox.functions.RetrieveActiveBridgedInterfaces.java  
  
+ protected <T> T propagate(Exception e) {  
+ Throwables.propagate(e);  
**+ assert false;**+ return null;  
+ }  
========================================

Query and catalog expect and live testing along with Checks and other cliemt and domain object fixes  
  
Project: jclouds.txt  
Commit: 973ef7e1672b16a2c1cf84ba06eacaef3049f673  
ClassName: org.jclouds.vcloud.director.v1\_5.domain.Checks.java  
  
@@ -123,5 +123,5 @@ public static void checkLink(Link link) {  
 **assertNotNull(link.getRel(), "The Rel attribute of a Link must be set");  
- assertTrue(Link.Rel.ALL.contains(link.getRel()),**- String.format("The Rel attribute (%s) of a Link must be one of the allowed list - %s",   
- link.getRel(), Iterables.toString(Link.Rel.ALL)));  
**+ assertTrue(Link.Rel.ALL.contains(link.getRel()), String.format("The Rel attribute of a Link must be from the allowed list: %s", Iterables.toString(Link.Rel.ALL)));  
+ assertTrue(Link.Rel.ALL.contains(link.getRel()), String.format("The Rel attribute of a Link cannot be '%s'", link.getRel()));**========================================

cosmetic fixes  
  
Project: jclouds.txt  
Commit: b0a9263486d579e412ecc2cf22894e81ecbb485c  
ClassName: org.jclouds.libvirt.compute.strategy.LibvirtComputeServiceAdapter.java  
  
+ protected <T> T propogate(LibvirtException e) {  
+ Throwables.propagate(e);  
**+ assert false;**+ return null;  
+ }  
+ protected <T> T propogate(Exception e) {  
+ Throwables.propagate(e);  
========================================

cosmetic fixes  
  
Project: jclouds.txt  
Commit: b0a9263486d579e412ecc2cf22894e81ecbb485c  
ClassName: org.jclouds.libvirt.compute.functions.DomainToHardware.java  
  
+ protected <T> T propagate(Exception e) {  
+ Throwables.propagate(e);  
**+ assert false;**+ return null;  
+ }  
 }  
\ No newline at end of file  
========================================

Issue 298: fixed regression on authorizationexception not propagating, and also missing gogrid file  
  
Project: jclouds.txt  
Commit: 9bac52d315212e93484f9bf98bd891a2cf50a21d  
ClassName: org.jclouds.util.Utils.java  
  
+ public static <T> T propagateAuthorizationOrOriginalException(Exception e) {  
+ AuthorizationException aex = getFirstThrowableOfType(e, AuthorizationException.class);  
+ if (aex != null)  
+ throw aex;  
+ Throwables.propagate(e);  
**+ assert false : "exception should have propogated " + e;**+ return null;  
+ }  
========================================

issue 384: IsoToMachine + Test started  
  
Project: jclouds.txt  
Commit: 1e2e41f7c1d1201407be4012fc23c5344a0b0706  
ClassName: org.jclouds.virtualbox.functions.IsoToIMachine.java  
  
+ protected ExecResponse runScriptOnNode(String nodeId, String command,  
+ RunScriptOptions options) {  
+ ExecResponse toReturn = context.getComputeService().runScriptOnNode(  
+ nodeId, command, options);  
**+ assert toReturn.getExitCode() == 0 : toReturn;**+ return toReturn;  
+ }  
========================================

fixed a minor vcloud bug and toString statements  
  
Project: jclouds.txt  
Commit: 5b300f87ccf8f1962800a2286f81d4f6127f8248  
ClassName: org.jclouds.vcloud.config.VCloudRestClientModule.java  
  
@@ -165,4 +165,5 @@ public VCloudSession get() {  
 } catch (Exception e) {  
- Throwables.propagateIfPossible(e);  
- throw new RuntimeException("Error logging in", e);  
+ Throwables.propagate(e);  
**+ assert false;**+ return null;  
 }  
========================================

issue 384: VirtualBoxComputeServiceAdapter  
  
Project: jclouds.txt  
Commit: c46c0b4c47b669b1b8f48aa0bf7ece9b0719ef8c  
ClassName: org.jclouds.virtualbox.compute.VirtualBoxComputeServiceAdapter.java  
  
+ protected <T> T propogate(Exception e) {  
+ Throwables.propagate(e);  
**+ assert false;**+ return null;  
 }  
========================================

Fixing some issues with XmlEnum annotated types  
  
Project: jclouds.txt  
Commit: 47cbbeef5783b88c15c95d6caa78288fddf698d5  
ClassName: org.jclouds.vcloud.director.v1\_5.domain.Checks.java  
  
@@ -1257,4 +1257,3 @@ private static void checkNetworkConnection(NetworkConnection val) {  
 **assertNotNull(val.getIpAddressAllocationMode(), String.format(NOT\_NULL\_OBJ\_FIELD\_FMT, "IpAddressAllocationMode", "NetworkConnection"));**- IpAddressAllocationMode mode = NetworkConnection.IpAddressAllocationMode.valueOf(val.getIpAddressAllocationMode());  
**- assertTrue(NetworkConnection.IpAddressAllocationMode.ALL.contains(mode),   
+ assertNotEquals(val.getIpAddressAllocationMode(), NetworkConnection.IpAddressAllocationMode.UNRECOGNIZED,** String.format(REQUIRED\_VALUE\_OBJECT\_FMT, "IpAddressAllocationMode", "NetworkConnection", val.getIpAddressAllocationMode(), Iterables.toString(NetworkConnection.IpAddressAllocationMode.ALL)));  
========================================

Changes to fix problems introduced by previous refactorings.  
  
Project: jsecurity.txt  
Commit: 2732bd6f8e6de2ecf0d6c2fda4100eb6a91e3e79  
ClassName: org.jsecurity.ri.authc.AbstractAuthenticator.java  
  
**- private void assertCreation( AuthorizationContext authzCtx ) throws IllegalStateException {  
+ private void assertCreation( SecurityContext authzCtx ) throws IllegalStateException {** if ( authzCtx == null ) {  
- String msg = "Programming or configuration error - No AuthorizationContext was created after successful " +  
+ String msg = "Programming or configuration error - No SecurityContext was created after successful " +  
 "authentication. Verify that you have either configured the " + getClass().getName() +  
- " instance with a proper " + AuthorizationContextFactory.class.getName() + " (easier) or " +  
+ " instance with a proper " + SecurityContextFactory.class.getName() + " (easier) or " +  
 "that you have overridden the " + AbstractAuthenticator.class.getName() +  
- ".createAuthorizationContext( AuthenticationInfo info ) method.";  
+ ".createSecurityContext( AuthenticationInfo info ) method.";  
 throw new IllegalStateException( msg );  
========================================

**Adjusted the UDDI WSDL to fix clashing ResponseWrapper names for getPublisherAssertions and setPublisherAssertions. Re-generated all the classes with wsimport.**  
Project: juddi.txt  
Commit: 6e90acf17c90e0dd8ede5c46f702502d3baeb47b  
ClassName: org.uddi.repl\_v3.ChangeRecord.java  
  
**+ public ChangeRecordPublisherAssertion getChangeRecordPublisherAssertion() {  
+ return changeRecordPublisherAssertion;**+ }  
========================================

**Adjusted the UDDI WSDL to fix clashing ResponseWrapper names for getPublisherAssertions and setPublisherAssertions. Re-generated all the classes with wsimport.**  
Project: juddi.txt  
Commit: 6e90acf17c90e0dd8ede5c46f702502d3baeb47b  
ClassName: org.uddi.api\_v3.ObjectFactory.java  
  
**+ public PublisherAssertionsResponse createPublisherAssertionsResponse() {  
+ return new PublisherAssertionsResponse();**+ }  
========================================

JUDDI-140, JUDDI-138: lots of work on validating user input and returning the appropriate error. Implemented messages that can be internationalized.  
  
Project: juddi.txt  
Commit: 06a2677d4299442199907ae1b41587e597347ef1  
ClassName: org.apache.juddi.validation.ValidatePublish.java  
  
**+ public static void validateDeletePublisherAssertions(EntityManager em, DeletePublisherAssertions body) throws DispositionReportFaultMessage {**+ if (body == null)  
+ throw new FatalErrorException(new ErrorMessage("errors.NullInput"));  
**+ List<org.uddi.api\_v3.PublisherAssertion> entityList = body.getPublisherAssertion();**+ if (entityList == null || entityList.size() == 0)  
**+ throw new AssertionNotFoundException(new ErrorMessage("errors.assertion.NoPubAssertions"));  
+ for (org.uddi.api\_v3.PublisherAssertion entity : entityList) {  
+ org.apache.juddi.model.PublisherAssertionId pubAssertionId = new org.apache.juddi.model.PublisherAssertionId(entity.getFromKey(), entity.getToKey());  
+ Object obj = em.find(org.apache.juddi.model.BusinessEntity.class, pubAssertionId);**+ if (obj == null)  
**+ throw new AssertionNotFoundException(new ErrorMessage("errors.assertion.AssertionNotFound", entity.getFromKey() + ", " + entity.getToKey()));**+ }  
+ }  
========================================

JUDDI-131: Started creating the layer to map api to model. Also added some basic infrastructure for config, error-handling, and the persistence framework. This is all likely to change, but was necessary to test my work.  
  
Project: juddi.txt  
Commit: fe7b3f04243a0376fd17f146d5b8a19526f19515  
ClassName: org.apache.juddi.mapping.MappingApiToModel.java  
  
**+ public static void mapPublisherAssertion(org.uddi.api\_v3.PublisherAssertion apiPubAssertion,   
+ org.apache.juddi.model.PublisherAssertion modelPubAssertion)**+ throws DispositionReportFaultMessage {  
**+ modelPubAssertion.setId(new org.apache.juddi.model.PublisherAssertionId(apiPubAssertion.getFromKey(), apiPubAssertion.getToKey()));**+ org.apache.juddi.model.BusinessEntity beFrom = new org.apache.juddi.model.BusinessEntity();  
**+ beFrom.setBusinessKey(apiPubAssertion.getFromKey());  
+ modelPubAssertion.setBusinessEntityByFromKey(beFrom);**+ org.apache.juddi.model.BusinessEntity beTo = new org.apache.juddi.model.BusinessEntity();  
**+ beFrom.setBusinessKey(apiPubAssertion.getToKey());  
+ modelPubAssertion.setBusinessEntityByToKey(beTo);  
+ org.uddi.api\_v3.KeyedReference apiKeyedRef = apiPubAssertion.getKeyedReference();**+ if (apiKeyedRef != null) {  
**+ modelPubAssertion.setTmodelKey(apiKeyedRef.getTModelKey());  
+ modelPubAssertion.setKeyName(apiKeyedRef.getKeyName());  
+ modelPubAssertion.setKeyValue(apiKeyedRef.getKeyName());**+ }  
+ }  
========================================

**Adjusted the UDDI WSDL to fix clashing ResponseWrapper names for getPublisherAssertions and setPublisherAssertions. Re-generated all the classes with wsimport.**  
Project: juddi.txt  
Commit: 6e90acf17c90e0dd8ede5c46f702502d3baeb47b  
ClassName: org.uddi.api\_v3.ObjectFactory.java  
  
**+ public JAXBElement<PublisherAssertion> createPublisherAssertion(PublisherAssertion value) {  
+ return new JAXBElement<PublisherAssertion>(\_PublisherAssertion\_QNAME, PublisherAssertion.class, null, value);**+ }  
========================================

**Adjusted the UDDI WSDL to fix clashing ResponseWrapper names for getPublisherAssertions and setPublisherAssertions. Re-generated all the classes with wsimport.**  
Project: juddi.txt  
Commit: 6e90acf17c90e0dd8ede5c46f702502d3baeb47b  
**ClassName: org.uddi.api\_v3.AddPublisherAssertions.java  
+ public List<PublisherAssertion> getPublisherAssertion() {  
+ if (publisherAssertion == null) {  
+ publisherAssertion = new ArrayList<PublisherAssertion>();**+ }  
**+ return this.publisherAssertion;**+ }  
========================================

**Adjusted the UDDI WSDL to fix clashing ResponseWrapper names for getPublisherAssertions and setPublisherAssertions. Re-generated all the classes with wsimport.**  
Project: juddi.txt  
Commit: 6e90acf17c90e0dd8ede5c46f702502d3baeb47b  
ClassName: org.uddi.repl\_v3.ChangeRecord.java  
  
**+ public void setChangeRecordPublisherAssertion(ChangeRecordPublisherAssertion value) {  
+ this.changeRecordPublisherAssertion = value;**+ }  
========================================

**Adjusted the UDDI WSDL to fix clashing ResponseWrapper names for getPublisherAssertions and setPublisherAssertions. Re-generated all the classes with wsimport.**  
Project: juddi.txt  
Commit: 6e90acf17c90e0dd8ede5c46f702502d3baeb47b  
ClassName: org.uddi.sub\_v3.SubscriptionResultsList.java  
  
**+ public AssertionStatusReport getAssertionStatusReport() {  
+ return assertionStatusReport;**+ }  
========================================

**Adjusted the UDDI WSDL to fix clashing ResponseWrapper names for getPublisherAssertions and setPublisherAssertions. Re-generated all the classes with wsimport.**  
Project: juddi.txt  
Commit: 6e90acf17c90e0dd8ede5c46f702502d3baeb47b  
ClassName: org.uddi.api\_v3.ObjectFactory.java  
  
**+ public AssertionStatusItem createAssertionStatusItem() {  
+ return new AssertionStatusItem();**+ }  
========================================

**Adjusted the UDDI WSDL to fix clashing ResponseWrapper names for getPublisherAssertions and setPublisherAssertions. Re-generated all the classes with wsimport.**  
Project: juddi.txt  
Commit: 6e90acf17c90e0dd8ede5c46f702502d3baeb47b  
ClassName: org.uddi.api\_v3.ObjectFactory.java  
  
**+ public JAXBElement<AddPublisherAssertions> createAddPublisherAssertions(AddPublisherAssertions value) {  
+ return new JAXBElement<AddPublisherAssertions>(\_AddPublisherAssertions\_QNAME, AddPublisherAssertions.class, null, value);**+ }  
========================================

SMX4KNL-104: When using a remote connection, errors are not reported correctly  
  
Project: karaf.txt  
Commit: 2ed67ca2fbb7f87f66b5dfcfa6ea6a8a28504f16  
ClassName: org.apache.geronimo.gshell.remote.client.proxy.SpringRemoteShellProxy.java  
  
+ private void displayError(final Throwable error) {  
**+ assert error != null;**+ Throwable cause = error;  
+ if (error instanceof ErrorNotification) {  
+ cause = error.getCause();  
+ }  
+ io.err.print("@|bold,red ERROR| ");  
+ io.err.print(cause.getClass().getSimpleName());  
+ io.err.println(": @|bold,red " + cause.getMessage() + "|");  
+ String stackTraceProperty = System.getProperty("gshell.show.stacktrace");  
+ boolean stackTraceFlag = false;  
+ if (stackTraceProperty != null) {  
+ stackTraceFlag = stackTraceProperty.trim().equals("true");  
+ }  
+ if (io.isDebug()) {  
+ log.debug(error.toString(), error);  
+ }  
+ else if (io.isVerbose() || stackTraceFlag) {  
+ StackTraceElement[] trace = cause.getStackTrace();  
+ StringBuffer buff = new StringBuffer();  
+ for (StackTraceElement e : trace) {  
+ buff.append(" @|bold at| ").  
+ append(e.getClassName()).  
  
+ append(".").  
+ append(e.getMethodName()).  
+ append(" (@|bold ");  
+ buff.append(e.isNativeMethod() ? "Native Method" :  
+ (e.getFileName() != null && e.getLineNumber() != -1 ? e.getFileName() + ":" + e.getLineNumber() :  
+ (e.getFileName() != null ? e.getFileName() : "Unknown Source")));  
+ buff.append("|)");  
+ io.err.println(buff);  
+ buff.setLength(0);  
+ }  
+ }  
+ }  
========================================

SMX4KNL-104: When using a remote connection, errors are not reported correctly  
  
Project: karaf.txt  
Commit: 2ed67ca2fbb7f87f66b5dfcfa6ea6a8a28504f16  
ClassName: org.apache.geronimo.gshell.remote.client.proxy.SpringRemoteShellProxy.java  
  
+ public void run(final Object... args) throws Exception {  
**+ assert args != null;**+ ensureOpened();  
+ log.debug("Starting interactive console; args: {}", args);  
+ final AtomicReference<ExitNotification> exitNotifHolder = new AtomicReference<ExitNotification>();  
+ final AtomicReference<Object> lastResultHolder = new AtomicReference<Object>();  
+ Console.Executor executor = new Console.Executor() {  
+ public Result execute(final String line) throws Exception {  
**+ assert line != null;**+ try {  
+ Object result = SpringRemoteShellProxy.this.execute(line);  
+ lastResultHolder.set(result);  
+ }  
+ catch (ExitNotification n) {  
+ exitNotifHolder.set(n);  
+ return Result.STOP;  
+ }  
+ return Result.CONTINUE;  
+ }  
+ };  
+ JLineConsole console = new JLineConsole(executor, io, terminal);  
+ console.setPrompter(new Console.Prompter() {  
+ Renderer renderer = new Renderer();  
+ public String prompt() {  
+ String userName = "user"; // shellInfo.getUserName();  
+ String hostName = "remote"; // shellInfo.getLocalHost().getHostName();  
+ String path = "/";  
+ return renderer.render("@|bold " + userName + "|@" + hostName + ":@|bold " + path + "|> ");  
+ }  
+ });  
+ console.setErrorHandler(new Console.ErrorHandler() {  
+ public Result handleError(final Throwable error) {  
**+ assert error != null;**+ displayError(error);  
+ return Result.CONTINUE;  
+ }  
+ });  
+ History history = new History();  
+ console.setHistory(history);  
+ console.setHistoryFile(new File(branding.getUserDirectory(), branding.getHistoryFileName()));  
+ if (!io.isQuiet()) {  
+ io.out.println(branding.getWelcomeBanner());  
+ }  
+ if (args.length != 0) {  
+ execute(args);  
+ }  
+ console.run();  
+ ExitNotification n = exitNotifHolder.get();  
+ if (n != null) {  
+ throw n;  
+ }  
+ }  
========================================

Another iteration of work toward KNOX-37 in the investigation of OAuth/OpenID for identity federation. Also fixed issues with provider resolution by name and various refactorings.  
  
Project: knox.txt  
Commit: 61ca72e10893fac097643aab490147408a306046  
**ClassName: org.apache.hadoop.gateway.filter.PseudoIdentityAssertionFilter.java**+ public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)   
+ throws IOException, ServletException {  
+ Subject subject = Subject.getSubject(AccessController.getContext());  
+ String principalName = getPrincipalName(subject);  
+ principalName = mapper.mapPrincipal(principalName);  
**+ IdentityAssertionHttpServletRequestWrapper wrapper =   
+ new IdentityAssertionHttpServletRequestWrapper(**+ (HttpServletRequest)request,   
+ principalName);  
+ chain.doFilter( wrapper, response );  
+ }  
========================================

Another iteration of work toward KNOX-37 in the investigation of OAuth/OpenID for identity federation. Also fixed issues with provider resolution by name and various refactorings.  
  
Project: knox.txt  
Commit: 61ca72e10893fac097643aab490147408a306046  
**ClassName: org.apache.hadoop.gateway.provider.federation.jwt.deploy.JWTAccessTokenAssertionContributor.java**+ public String getName() {  
**+ return "JWTAccessTokenAsserter";**+ }  
========================================

Another iteration of work toward KNOX-37 in the investigation of OAuth/OpenID for identity federation. Also fixed issues with provider resolution by name and various refactorings.  
  
Project: knox.txt  
Commit: 61ca72e10893fac097643aab490147408a306046  
**ClassName: org.apache.hadoop.gateway.provider.federation.jwt.deploy.JWTAuthCodeAssertionContributor.java**+ public String getRole() {  
**+ return "identity-assertion";**+ }  
========================================

Another iteration of work toward KNOX-37 in the investigation of OAuth/OpenID for identity federation. Also fixed issues with provider resolution by name and various refactorings.  
  
Project: knox.txt  
Commit: 61ca72e10893fac097643aab490147408a306046  
**ClassName: org.apache.hadoop.gateway.provider.federation.jwt.deploy.JWTAuthCodeAssertionContributor.java**+ public String getName() {  
**+ return "JWTAuthCodeAsserter";**+ }  
========================================

Another iteration of work toward KNOX-37 in the investigation of OAuth/OpenID for identity federation. Also fixed issues with provider resolution by name and various refactorings.  
  
Project: knox.txt  
Commit: 61ca72e10893fac097643aab490147408a306046  
**ClassName: org.apache.hadoop.gateway.provider.federation.jwt.deploy.JWTAccessTokenAssertionContributor.java**+ public String getRole() {  
**+ return "identity-assertion";**+ }  
========================================

a lot of compile errors still, but I'm counting on you guys :-)  
  
Project: kylin.txt  
Commit: 2fd289f9552ab91660fa758b9346e5af68d4e509  
ClassName: com.kylinolap.cube.model.CubeDesc.java  
  
+ public TblColRef findPKByFK(TblColRef fk) {  
**+ assert isFactTable(fk.getTable());**+ TblColRef candidate = null;  
+ for (DimensionDesc dim : dimensions) {  
+ JoinDesc join = dim.getJoin();  
+ if (join == null)  
+ continue;  
+ int find = ArrayUtils.indexOf(join.getForeignKeyColumns(), fk);  
+ if (find >= 0) {  
+ candidate = join.getPrimaryKeyColumns()[find];  
+ if (join.getForeignKeyColumns().length == 1) { // is single  
+ break;  
+ }  
+ }  
+ }  
+ return candidate;  
+ }  
========================================

fix #76  
  
Project: kylin.txt  
Commit: e1a96c0be23617ccada20daa1645b57626d399a7  
ClassName: com.kylinolap.job.hadoop.cube.NewBaseCuboidMapper.java  
  
@@ -266,3 +265,3 @@ private void prepareMetrics() {  
**- Assert.isTrue(filledDimension == keyBytesBuf.length);  
+ assert filledDimension == keyBytesBuf.length;**========================================

fix compile issue  
  
Project: kylin.txt  
Commit: 54f8a30b24c30e86b0f4c711682e9b7a00f3af09  
ClassName: com.kylinolap.rest.service.CubeService.java  
  
+ private boolean isCubeInProject(String projectName, CubeInstance target) {  
+ ProjectManager projectManager = getProjectManager();  
+ ProjectInstance project = projectManager.getProject(projectName);  
+ if (project == null) {  
+ return false;  
+ }  
+ for (ProjectDataModel projectDataModel: project.getDataModels()) {  
+ if (projectDataModel.getType() == DataModelRealizationType.CUBE) {  
+ CubeInstance cube = getCubeManager().getCube(projectDataModel.getRealization());  
**+ assert cube != null;**+ if (cube.equals(target)) {  
+ return true;  
+ }  
+ }  
+ }  
+ return false;  
+ }  
========================================

better error handling  
  
Project: kylin.txt  
Commit: 5fb51ffadc6d5841f6f2b94c19abd305ec6fe34e  
ClassName: com.kylinolap.rest.controller.QueryController.java  
  
@@ -214,7 +218,6 @@ private SQLResponse doQuery(SQLRequest sqlRequest) {  
 } catch (AccessDeniedException ade) {  
 logger.error("Exception when execute sql", ade);  
 throw new ForbiddenException(ade.getLocalizedMessage());  
- } catch (Exception e) {  
**+ } catch (Throwable e) { // calcite may throw AssertError** SQLResponse exceptionRes = new SQLResponse(null, null, 0, true, e.getMessage());  
========================================

fixed len metrics done  
  
Project: kylin.txt  
Commit: 13eaee0a3311bec5fdd3983a0aeca09ef8fc22dd  
ClassName: com.kylinolap.metadata.model.invertedindex.InvertedIndexDesc.java  
  
+ public boolean isMetricsCol(TblColRef col) {  
**+ assert col.getTable().equals(factTable);**+ return isMetricsCol(col.getColumn().getZeroBasedIndex());  
+ }  
+ public boolean isMetricsCol(int colZeroBasedIndex) {  
+ return metricsColSet.get(colZeroBasedIndex);  
========================================

MAHOUT-593: initial addition of Stochastic SVD method (related issue is MAHOUT-376)  
  
Project: mahout.txt  
Commit: 151de0d737501af5dcfee8a21bc7d18ff6edddc8  
ClassName: org.apache.mahout.math.hadoop.stochasticsvd.GivensThinSolver.java  
  
+ public void solve(Matrix a) {  
**+ assert a.rowSize() == m;  
+ assert a.columnSize() == n;**+ double[] aRow = new double[n];  
+ for (int i = 0; i < m; i++) {  
+ Vector aRowV = a.getRow(i);  
+ for (int j = 0; j < n; j++)  
+ aRow[j] = aRowV.getQuick(j);  
+ appendRow(aRow);  
+ }  
+ }  
========================================

MAHOUT-593: initial addition of Stochastic SVD method (related issue is MAHOUT-376)  
  
Project: mahout.txt  
Commit: 151de0d737501af5dcfee8a21bc7d18ff6edddc8  
ClassName: org.apache.mahout.math.hadoop.stochasticsvd.GivensThinSolver.java  
  
+ public static void mergeR(UpperTriangular r1, UpperTriangular r2) {  
+ TriangularRowView r1Row = new TriangularRowView(r1);  
+ TriangularRowView r2Row = new TriangularRowView(r2);  
+ int kp = r1Row.size();  
**+ assert kp == r2Row.size();**+ double[] cs = new double[2];  
+ for (int v = 0; v < kp; v++) {  
+ for (int u = v; u < kp; u++) {  
+ givens(r1Row.setViewedRow(u).get(u), r2Row.setViewedRow(u - v).get(u),  
+ cs);  
+ applyGivensInPlace(cs[0], cs[1], r1Row, r2Row, u, kp - u);  
+ }  
+ }  
+ }  
========================================

MAHOUT-593: initial addition of Stochastic SVD method (related issue is MAHOUT-376)  
  
Project: mahout.txt  
Commit: 151de0d737501af5dcfee8a21bc7d18ff6edddc8  
ClassName: org.apache.mahout.math.hadoop.stochasticsvd.GivensThinSolver.java  
  
+ public static void mergeRonQ(double[][] r1, double[][] r2, double[][] qt1,  
+ double[][] qt2) {  
+ int kp = r1[0].length;  
**+ assert kp == r2[0].length;  
+ assert kp == qt1.length;  
+ assert kp == qt2.length;**+ int r = qt1[0].length;  
**+ assert qt2[0].length == r;**+ double[] cs = new double[2];  
+ for (int v = 0; v < kp; v++) {  
+ for (int u = v; u < kp; u++) {  
+ givens(r1[u][u], r2[u - v][u], cs);  
+ applyGivensInPlace(cs[0], cs[1], r1[u], r2[u - v], u, kp - u);  
+ applyGivensInPlace(cs[0], cs[1], qt1[u], qt2[u - v], 0, r);  
+ }  
+ }  
+ }  
========================================

MAHOUT-593: initial addition of Stochastic SVD method (related issue is MAHOUT-376)  
  
Project: mahout.txt  
Commit: 151de0d737501af5dcfee8a21bc7d18ff6edddc8  
ClassName: org.apache.mahout.math.hadoop.stochasticsvd.SSVDPrototype.java  
  
+ public static void testBlockQrWithSSVD(int dims, int kp, int r,  
+ final long rndSeed) throws Exception {  
+ DenseMatrix mx = new DenseMatrix(dims << 2, dims);  
+ Random rnd = new Random();  
+ for (int i = 0; i < mx.rowSize(); i++)  
+ for (int j = 0; j < mx.columnSize(); j++)  
+ mx.set(i, j, (rnd.nextDouble() - 0.5) \* 1000);  
+ mx.setQuick(0, 0, 1);  
+ mx.setQuick(0, 1, 2);  
+ mx.setQuick(0, 2, 3);  
+ mx.setQuick(1, 0, 4);  
+ mx.setQuick(1, 1, 5);  
+ mx.setQuick(1, 2, 6);  
+ mx.setQuick(2, 0, 7);  
+ mx.setQuick(2, 1, 8);  
+ mx.setQuick(2, 2, 9);  
+ SingularValueDecomposition svd2 = new SingularValueDecomposition(mx);  
+ double[] svaluesControl = svd2.getSingularValues();  
+ for (int i = 0; i < kp; i++)  
+ System.out.printf("%e ", svaluesControl[i]);  
+ System.out.println();  
+ int m = mx.rowSize(); /\* ,n=mx.columnSize(); \*/  
+ final Map<Integer, Vector> btRows = new HashMap<Integer, Vector>();  
+ PartialRowEmitter btEmitter = new PartialRowEmitter() {  
+ @Override  
+ public void emitRow(int rowNum, Vector row) throws IOException {  
+ Vector btRow = btRows.get(rowNum);  
+ if (btRow != null)  
+ row.addTo(btRow);  
+ btRows.put(rowNum, btRow == null ? new DenseVector(row) : btRow);  
+ }  
+ };  
+ SSVDPrototype mapperSimulation = new SSVDPrototype(rndSeed, kp, r);  
+ for (int i = 0; i < m; i++)  
+ mapperSimulation.firstPass(i, mx.getRow(i));  
+ mapperSimulation.finishFirstPass();  
+ for (int i = 0; i < m; i++)  
+ mapperSimulation.secondPass(i, mx.getRow(i), btEmitter);  
+ final Map<Integer, Vector> bbt = new HashMap<Integer, Vector>();  
+ PartialRowEmitter bbtEmitter = new PartialRowEmitter() {  
+ @Override  
+ public void emitRow(int rowNum, Vector row) throws IOException {  
+ Vector bbtRow = bbt.get(rowNum);  
+ if (bbtRow != null)  
+ row.addTo(bbtRow);  
+ bbt.put(rowNum, bbtRow == null ? new DenseVector(row) : bbtRow);  
+ }  
+ };  
+ for (Map.Entry<Integer, Vector> btRowEntry : btRows.entrySet()) {  
+ Vector btRow = btRowEntry.getValue();  
**+ assert btRow.size() == kp;**+ for (int i = 0; i < kp; i++)  
+ bbtEmitter.emitRow(i, btRow.times(btRow.getQuick(i)));  
+ }  
+ double[][] bbtValues = new double[kp][];  
+ for (int i = 0; i < kp; i++) {  
+ bbtValues[i] = new double[kp];  
+ Vector bbtRow = bbt.get(i);  
+ for (int j = 0; j < kp; j++)  
+ bbtValues[i][j] = bbtRow.getQuick(j);  
+ }  
+ EigenSolverWrapper eigenWrapper = new EigenSolverWrapper(bbtValues);  
+ double[] eigenva2 = eigenWrapper.getEigenValues();  
+ double[] svalues = new double[kp];  
+ for (int i = 0; i < kp; i++)  
+ svalues[i] = Math.sqrt(eigenva2[i]); // sqrt?  
+ for (int i = 0; i < kp; i++)  
+ System.out.printf("%e ", svalues[i]);  
+ System.out.println();  
+ }  
========================================

MAHOUT-593: initial addition of Stochastic SVD method (related issue is MAHOUT-376)  
  
Project: mahout.txt  
Commit: 151de0d737501af5dcfee8a21bc7d18ff6edddc8  
ClassName: org.apache.mahout.math.hadoop.stochasticsvd.QJob.java  
  
+ protected void cleanup(Context context) throws IOException,  
+ InterruptedException {  
+ try {  
+ if (qSolver == null && yLookahead.size() == 0)  
+ return;  
+ if (qSolver == null)  
+ qSolver = new GivensThinSolver(yLookahead.size(), kp);  
+ qSolver.adjust(qSolver.getCnt() + yLookahead.size());  
+ while (yLookahead.size() > 0) {  
+ qSolver.appendRow(yLookahead.remove(0));  
+ }  
**+ assert qSolver.isFull();**+ if (++blockCnt > 1) {  
+ flushSolver(context);  
**+ assert tempQw != null;**+ closeables.remove(tempQw);  
+ tempQw.close();  
+ }  
+ flushQBlocks(context);  
+ } finally {  
+ IOUtils.close(closeables);  
+ }  
+ }  
========================================

MAHOUT-593: initial addition of Stochastic SVD method (related issue is MAHOUT-376)  
  
Project: mahout.txt  
Commit: 151de0d737501af5dcfee8a21bc7d18ff6edddc8  
ClassName: org.apache.mahout.math.hadoop.stochasticsvd.GivensThinSolver.java  
  
+ public static void mergeRonQ(UpperTriangular r1, UpperTriangular r2,  
+ double[][] qt1, double[][] qt2) {  
+ TriangularRowView r1Row = new TriangularRowView(r1);  
+ TriangularRowView r2Row = new TriangularRowView(r2);  
+ int kp = r1Row.size();  
**+ assert kp == r2Row.size();  
+ assert kp == qt1.length;  
+ assert kp == qt2.length;**+ int r = qt1[0].length;  
**+ assert qt2[0].length == r;**+ double[] cs = new double[2];  
+ for (int v = 0; v < kp; v++) {  
+ for (int u = v; u < kp; u++) {  
+ givens(r1Row.setViewedRow(u).get(u), r2Row.setViewedRow(u - v).get(u),  
+ cs);  
+ applyGivensInPlace(cs[0], cs[1], r1Row, r2Row, u, kp - u);  
+ applyGivensInPlace(cs[0], cs[1], qt1[u], qt2[u - v], 0, r);  
+ }  
+ }  
+ }  
========================================

MAHOUT-593: initial addition of Stochastic SVD method (related issue is MAHOUT-376)  
  
Project: mahout.txt  
Commit: 151de0d737501af5dcfee8a21bc7d18ff6edddc8  
ClassName: org.apache.mahout.math.hadoop.stochasticsvd.GivensThinSolver.java  
  
+ public static void mergeR(double[][] r1, double[][] r2) {  
+ int kp = r1[0].length;  
**+ assert kp == r2[0].length;**+ double[] cs = new double[2];  
+ for (int v = 0; v < kp; v++) {  
+ for (int u = v; u < kp; u++) {  
+ givens(r1[u][u], r2[u - v][u], cs);  
+ applyGivensInPlace(cs[0], cs[1], r1[u], r2[u - v], u, kp - u);  
+ }  
+ }  
+ }  
========================================

MAHOUT-593: initial addition of Stochastic SVD method (related issue is MAHOUT-376)  
  
Project: mahout.txt  
Commit: 151de0d737501af5dcfee8a21bc7d18ff6edddc8  
ClassName: org.apache.mahout.math.hadoop.stochasticsvd.QJob.java  
  
+ private void secondPass(Context ctx) throws IOException,  
+ InterruptedException {  
+ qSolver = null; // release mem  
+ FileSystem localFs = FileSystem.getLocal(ctx.getConfiguration());  
+ SequenceFile.Reader tempQr = new SequenceFile.Reader(localFs,  
+ tempQPath, ctx.getConfiguration());  
+ closeables.addFirst(tempQr);  
+ int qCnt = 0;  
+ while (tempQr.next(tempKey, value)) {  
+ value  
+ .setBlock(GivensThinSolver.computeQtHat(value.getBlock(), qCnt,  
+ new GivensThinSolver.DeepCopyUTIterator(rSubseq.iterator())));  
+ if (qCnt == 1) // just merge r[0] <- r[1] so it doesn't have to repeat  
+ GivensThinSolver.mergeR(rSubseq.get(0), rSubseq.remove(1));  
+ else  
+ qCnt++;  
+ outputs.getCollector(OUTPUT\_QHAT, null).collect(key, value);  
+ }  
**+ assert rSubseq.size() == 1;**+ outputs.getCollector(OUTPUT\_R, null)  
+ .collect(  
+ key,  
+ new VectorWritable(new DenseVector(rSubseq.get(0).getData(),  
+ true)));  
+ }  
========================================

MAHOUT-593: initial addition of Stochastic SVD method (related issue is MAHOUT-376)  
  
Project: mahout.txt  
Commit: 151de0d737501af5dcfee8a21bc7d18ff6edddc8  
ClassName: org.apache.mahout.math.hadoop.stochasticsvd.Omega.java  
  
+ public void computeYRow(Vector aRow, double[] yRow) {  
**+ assert yRow.length == kp;**+ Arrays.fill(yRow, 0);  
+ if (aRow instanceof SequentialAccessSparseVector) {  
+ int j = 0;  
+ for (Element el : aRow) {  
+ accumDots(j, el.get(), yRow);  
+ j++;  
+ }  
+ } else {  
+ int n = aRow.size();  
+ for (int j = 0; j < n; j++)  
+ accumDots(j, aRow.getQuick(j), yRow);  
+ }  
+ }  
========================================

MAHOUT-593: initial addition of Stochastic SVD method (related issue is MAHOUT-376)  
  
Project: mahout.txt  
Commit: 151de0d737501af5dcfee8a21bc7d18ff6edddc8  
ClassName: org.apache.mahout.math.hadoop.stochasticsvd.SSVDSolver.java  
  
+ public void run() throws IOException {  
+ LinkedList<Closeable> closeables = new LinkedList<Closeable>();  
+ try {  
+ Class<? extends Writable> labelType = sniffInputLabelType(inputPath,  
+ conf, closeables);  
+ FileSystem fs = FileSystem.get(conf);  
+ Path qPath = new Path(outputPath, "Q-job");  
+ Path btPath = new Path(outputPath, "Bt-job");  
+ Path bbtPath = new Path(outputPath, "BBt-job");  
+ Path uHatPath = new Path(outputPath, "UHat");  
+ Path svPath = new Path(outputPath, "Sigma");  
+ Path uPath = new Path(outputPath, "U");  
+ Path vPath = new Path(outputPath, "V");  
+ fs.delete(qPath, true); // or we can't re-run it repeatedly, just in case.  
+ fs.delete(btPath, true);  
+ fs.delete(bbtPath, true);  
+ fs.delete(uHatPath, true);  
+ fs.delete(svPath, true);  
+ fs.delete(uPath, true);  
+ fs.delete(vPath, true);  
+ Random rnd = new Random();  
+ long seed = rnd.nextLong();  
+ QJob.run(conf, inputPath, qPath, ablockRows, minSplitSize, k, p, seed,  
+ reduceTasks);  
+ BtJob.run(conf, inputPath, qPath, btPath, minSplitSize, k, p,  
+ reduceTasks, labelType);  
+ BBtJob.run(conf, new Path(btPath, BtJob.OUTPUT\_BT + "-\*"), bbtPath, 1);  
+ UpperTriangular bbt = loadUpperTriangularMatrix(fs, new Path(bbtPath,  
+ BBtJob.OUTPUT\_BBT + "-\*"), conf);  
**+ assert bbt.columnSize() == k + p;**+ double[][] bbtSquare = new double[k + p][];  
+ for (int i = 0; i < k + p; i++)  
+ bbtSquare[i] = new double[k + p];  
+ for (int i = 0; i < k + p; i++)  
+ for (int j = i; j < k + p; j++)  
+ bbtSquare[i][j] = bbtSquare[j][i] = bbt.getQuick(i, j);  
+ svalues = new double[k + p];  
+ EigenSolverWrapper eigenWrapper = new EigenSolverWrapper(bbtSquare);  
+ double[] eigenva2 = eigenWrapper.getEigenValues();  
+ for (int i = 0; i < k + p; i++)  
+ svalues[i] = Math.sqrt(eigenva2[i]); // sqrt?  
+ double[][] uHat = eigenWrapper.getUHat();  
+ fs.mkdirs(uHatPath);  
+ SequenceFile.Writer uHatWriter = SequenceFile.createWriter(fs, conf,  
+ uHatPath = new Path(uHatPath, "uhat.seq"), IntWritable.class,  
+ VectorWritable.class, CompressionType.BLOCK);  
+ closeables.addFirst(uHatWriter);  
+ int m = uHat.length;  
+ IntWritable iw = new IntWritable();  
+ VectorWritable vw = new VectorWritable();  
+ for (int i = 0; i < m; i++) {  
+ vw.set(new DenseVector(uHat[i], true));  
+ iw.set(i);  
+ uHatWriter.append(iw, vw);  
+ }  
+ closeables.remove(uHatWriter);  
+ uHatWriter.close();  
+ SequenceFile.Writer svWriter = SequenceFile.createWriter(fs, conf,  
+ svPath = new Path(svPath, "svalues.seq"), IntWritable.class,  
+ VectorWritable.class, CompressionType.BLOCK);  
+ closeables.addFirst(svWriter);  
+ vw.set(new DenseVector(svalues, true));  
+ svWriter.append(iw, vw);  
+ closeables.remove(svWriter);  
+ svWriter.close();  
+ UJob ujob = null;  
+ VJob vjob = null;  
+ if (computeU)  
+ (ujob = new UJob()).start(conf,  
+ new Path(btPath, BtJob.OUTPUT\_Q + "-\*"), uHatPath, svPath, uPath,  
+ k, reduceTasks, labelType, cUHalfSigma); // actually this is  
+ if (computeV)  
+ (vjob = new VJob()).start(conf,  
+ new Path(btPath, BtJob.OUTPUT\_BT + "-\*"), uHatPath, svPath, vPath,  
+ k, reduceTasks, cVHalfSigma);  
+ if (ujob != null) {  
+ ujob.waitForCompletion();  
+ this.uPath = uPath.toString();  
+ }  
+ if (vjob != null) {  
+ vjob.waitForCompletion();  
+ this.vPath = vPath.toString();  
+ }  
+ } catch (InterruptedException exc) {  
+ throw new IOException("Interrupted", exc);  
+ } catch (ClassNotFoundException exc) {  
+ throw new IOException(exc);  
+ } finally {  
+ IOUtils.close(closeables);  
+ }  
+ }  
========================================

Fix for CONNECTORS-146. Introduce a new flow for documents being cleaned up, as opposed to being deleted as part of a job's deletion. This furthermore allows us to fix a problem with carrydown data being affected by document cleanup.  
  
Project: manifoldcf.txt  
Commit: f6122c1cc2babd912fb8c42777b80a28b9b9608a  
ClassName: org.apache.manifoldcf.crawler.jobs.JobManager.java  
  
@@ -2489,2 +2686,93 @@ public void resetDocumentMultiple(DocumentDescription[] documentDescriptions, lo  
+ public void resetCleaningDocumentMultiple(DocumentDescription[] documentDescriptions)  
+ throws ManifoldCFException  
+ {  
+ Long[] ids = new Long[documentDescriptions.length];  
+ String[] docIDHashes = new String[documentDescriptions.length];  
+ HashMap indexMap = new HashMap();  
+ int i = 0;  
+ while (i < documentDescriptions.length)  
+ {  
+ docIDHashes[i] =documentDescriptions[i].getDocumentIdentifierHash() + ":" + documentDescriptions[i].getJobID();  
+ indexMap.put(docIDHashes[i],new Integer(i));  
+ i++;  
+ }  
+ java.util.Arrays.sort(docIDHashes);  
+ i = 0;  
+ while (i < docIDHashes.length)  
+ {  
+ String docIDHash = docIDHashes[i];  
+ Integer x = (Integer)indexMap.remove(docIDHash);  
+ if (x == null)  
**+ throw new ManifoldCFException("Assertion failure: duplicate document identifier jobid/hash detected!");**+ int index = x.intValue();  
+ ids[i] = documentDescriptions[index].getID();  
+ i++;  
+ }  
+ while (true)  
+ {  
+ long sleepAmt = 0L;  
+ database.beginTransaction();  
+ try  
+ {  
+ i = 0;  
+ while (i < ids.length)  
+ {  
+ jobQueue.setUncleaningStatus(ids[i]);  
+ i++;  
+ }  
+ break;  
+ }  
+ catch (ManifoldCFException e)  
+ {  
+ database.signalRollback();  
+ if (e.getErrorCode() == e.DATABASE\_TRANSACTION\_ABORT)  
+ {  
+ if (Logging.perf.isDebugEnabled())  
+ Logging.perf.debug("Aborted transaction resetting cleaning documents: "+e.getMessage());  
+ sleepAmt = getRandomAmount();  
+ continue;  
+ }  
+ throw e;  
+ }  
+ catch (Error e)  
+ {  
+ database.signalRollback();  
+ throw e;  
+ }  
+ finally  
+ {  
+ database.endTransaction();  
+ sleepFor(sleepAmt);  
+ }  
+ }  
+ }  
========================================

SOLR-2089: fixed tests + exception when facet.offset is beyond queue size  
  
Project: manifoldcf-integration-solr-3.x.txt  
Commit: c11fce303f40f0fa34883bbb8019cd62c87e4044  
ClassName: org.apache.solr.request.UnInvertedField.java  
  
@@ -585,4 +585,6 @@ public NamedList getCounts(SolrIndexSearcher searcher, DocSet baseDocs, int offs  
- final int[] tnums = new int[Math.min(queue.size()-off, lim)];  
+ final int[] tnums = new int[Math.min(Math.max(0, queue.size()-off), lim)];  
 final int[] indirect = counts; // reuse the counts array for the index into the tnums array  
**+ assert indirect.length >= tnums.length;** int tnumCount = 0;  
========================================

SOLR-1297: fix sort by function parsing (merge 1003107 and part of 997870,998707(JSONTestUtil.java) to pass TestFunctionQuery)  
  
Project: manifoldcf-integration-solr-3.x.txt  
Commit: 130f5b51b52b578858a2db4e152a6336254fb5a9  
ClassName: org.apache.solr.search.QueryParsing.java  
  
- public static Sort parseSort(String sortSpec, IndexSchema schema) {  
+ public static Sort parseSort(String sortSpec, SolrQueryRequest req) {  
 if (sortSpec == null || sortSpec.length() == 0) return null;  
- char[] chars = sortSpec.toCharArray();  
- int i = 0;  
- StringBuilder buffer = new StringBuilder(sortSpec.length());  
- String sort = null;  
- String order = null;  
- int functionDepth = 0;  
- boolean score = true;  
- List<SortField> lst = new ArrayList<SortField>(5);  
- boolean needOrder = false;  
- while (i < chars.length) {  
- if (Character.isWhitespace(chars[i]) && functionDepth == 0) {  
- if (buffer.length() == 0) {  
- } else {  
- if (needOrder == false) {  
- sort = buffer.toString().trim();  
- buffer.setLength(0);  
- needOrder = true;  
- } else {  
- order = buffer.toString().trim();  
- buffer.setLength(0);  
- needOrder = false;  
- }  
- }  
- } else if (chars[i] == '(' && functionDepth >= 0) {  
- buffer.append(chars[i]);  
- functionDepth++;  
- } else if (chars[i] == ')' && functionDepth > 0) {  
- buffer.append(chars[i]);  
- functionDepth--;//close up one layer  
- } else if (chars[i] == ',' && functionDepth == 0) {//can either be a separator of sort declarations, or a separator in a function  
- if (needOrder == true && buffer.length() > 0){  
- order = buffer.toString().trim();  
- buffer.setLength(0);  
- needOrder = false;  
- }  
- score = processSort(schema, sort, order, lst);  
- sort = null;  
- order = null;  
- buffer.setLength(0);//get ready for the next one, if there is one  
- } else if (chars[i] == ',' && functionDepth > 0) {  
- buffer.append(chars[i]);  
+ List<SortField> lst = new ArrayList<SortField>(4);  
+ try {  
+ StrParser sp = new StrParser(sortSpec);  
+ while (sp.pos < sp.end) {  
+ sp.eatws();  
+ int start = sp.pos;  
+ String field = sp.getId(null);  
+ ValueSource vs = null;  
+ if (field == null || sp.ch() != ' ') {  
+ String funcStr = sp.val.substring(start);  
+ QParser parser = QParser.getParser(funcStr, FunctionQParserPlugin.NAME, req);  
+ Query q = null;  
+ if (parser instanceof FunctionQParser) {  
+ FunctionQParser fparser = (FunctionQParser)parser;  
+ fparser.setParseMultipleSources(false);  
+ fparser.setParseToEnd(false);  
+ q = fparser.getQuery();  
+ if (fparser.localParams != null) {  
+ if (fparser.valFollowedParams) {  
+ int leftOver = fparser.sp.end - fparser.sp.pos;  
+ sp.pos = sp.end - leftOver; // reset our parser to the same amount of leftover  
 } else {  
- buffer.append(chars[i]);  
+ sp.pos = start + fparser.localParamsEnd;  
 }  
- i++;  
- }  
- if (buffer.length() > 0 && needOrder){//see if we have anything left, at most it should be an order  
- order = buffer.toString().trim();  
- buffer.setLength(0);  
- needOrder = false;  
+ } else {  
+ int leftOver = fparser.sp.end - fparser.sp.pos;  
+ sp.pos = sp.end - leftOver; // reset our parser to the same amount of leftover  
 }  
+ } else {  
+ q = parser.getQuery();  
- if (functionDepth != 0){  
- throw new SolrException(SolrException.ErrorCode.BAD\_REQUEST, "Unable to parse sort spec, mismatched parentheses: " + sortSpec);  
- }  
- if (buffer.length() > 0){//there's something wrong, as everything should have been parsed by now  
- throw new SolrException(SolrException.ErrorCode.BAD\_REQUEST, "Unable to parse sort spec: " + sortSpec);  
**+ assert parser.getLocalParams() != null;**+ sp.pos = start + parser.localParamsEnd;  
 }  
  
- score = processSort(schema, sort, order, lst);  
- }  
- if (lst.size() == 1 && score == true && lst.get(0).getReverse() == false) {  
- return null; // do normal scoring...  
+ if (q instanceof FunctionQuery) {  
+ vs = ((FunctionQuery)q).getValueSource();  
+ } else {  
+ vs = new QueryValueSource(q, 0.0f);  
 }  
- return new Sort((SortField[]) lst.toArray(new SortField[lst.size()]));  
 }  
- private static boolean processSort(IndexSchema schema, String sort, String order, List<SortField> lst) {  
- boolean score = false;  
- if (sort != null && order != null) {  
- boolean top = true;  
+ String order = sp.getId("Expected sort order asc/desc");  
+ boolean top;  
 if ("desc".equals(order) || "top".equals(order)) {  
========================================

LUCENE-2593 (on 3x): fix certain rare cases where disk full could cause index corruption  
  
Project: manifoldcf-integration-solr-3.x.txt  
Commit: a27b2c434adbece0e04c1a91b1c6163f0097ad98  
ClassName: org.apache.lucene.index.IndexWriter.java  
  
 synchronized void commit() throws IOException {  
**+ assert Thread.holdsLock(IndexWriter.this);** for (Map.Entry<SegmentInfo,SegmentReader> ent : readerMap.entrySet()) {  
========================================

LUCENE-2593 (on 3x): fix certain rare cases where disk full could cause index corruption  
  
Project: manifoldcf-integration-solr-3.x.txt  
Commit: a27b2c434adbece0e04c1a91b1c6163f0097ad98  
ClassName: org.apache.lucene.index.IndexWriter.java  
  
@@ -489,3 +489,3 @@ public synchronized void release(SegmentReader sr, boolean drop) throws IOExcept  
**- assert !pooled | readerMap.get(sr.getSegmentInfo()) == sr;  
+ assert !pooled || readerMap.get(sr.getSegmentInfo()) == sr;**========================================

LUCENE-2576: simplify IndexWriter's private startCommit method now that it's single thread'd (may fix this intermittent test failure); also add one missing checkpoint() in addIndexes  
  
Project: manifoldcf-integration-solr-3.x.txt  
Commit: da87ec6fc53cf3a8aeac6f13005b28021e4b0ccc  
ClassName: org.apache.lucene.index.IndexWriter.java  
  
@@ -4357,21 +4364,6 @@ private void startCommit(long sizeInBytes, Map<String,String> commitUserData) th  
 synchronized(this) {  
- while(true) {  
- if (myChangeCount <= lastCommitChangeCount) {  
- if (infoStream != null) {  
- message("sync superseded by newer infos");  
- }  
- break;  
- } else if (pendingCommit == null) {  
- if (segmentInfos.getGeneration() > toSync.getGeneration())  
- toSync.updateGeneration(segmentInfos);  
**+ assert pendingCommit == null;**- boolean success = false;  
- try {  
**+ assert segmentInfos.getGeneration() == toSync.getGeneration();**========================================

LUCENE-2593 (on 3x): fix certain rare cases where disk full could cause index corruption  
  
Project: manifoldcf-integration-solr-3.x.txt  
Commit: a27b2c434adbece0e04c1a91b1c6163f0097ad98  
ClassName: org.apache.lucene.index.IndexWriter.java  
  
@@ -4352,2 +4337,9 @@ private void startCommit(long sizeInBytes, Map<String,String> commitUserData) th  
 **assert directory.fileExists(fileName): "file " + fileName + " does not exist";  
+ assert deleter.exists(fileName);** }  
========================================

LUCENE-2811: fix bug in IndexWriter.addIndexes(Directory[]) caused by SI.hasVector  
  
Project: manifoldcf-integration-solr-3.x.txt  
Commit: c9ebe502af4a0fdf624bbe1169c34cea9babe504  
ClassName: org.apache.lucene.index.IndexWriter.java  
  
@@ -3039,6 +3042,14 @@ public void addIndexes(Directory... dirs) throws CorruptIndexException, IOExcept  
 for (String file: info.files()) {  
- if (docStoreCopied && IndexFileNames.isDocStoreFile(file)) {  
+ final String newFileName;  
+ if (IndexFileNames.isDocStoreFile(file)) {  
+ newFileName = newDsName + IndexFileNames.stripSegmentName(file);  
+ if (dsFilesCopied.contains(newFileName)) {  
 continue;  
 }  
- dir.copy(directory, file, newSegName + IndexFileNames.stripSegmentName(file));  
+ dsFilesCopied.add(newFileName);  
+ } else {  
+ newFileName = newSegName + IndexFileNames.stripSegmentName(file);  
+ }  
**+ assert !directory.fileExists(newFileName): "file \"" + newFileName + "\" already exists";**+ dir.copy(directory, file, newFileName);  
 }  
========================================

LUCENE-2295: Added a LimitTokenCountAnalyzer / LimitTokenCountFilter to wrap any other Analyzer and provide the same functionality as MaxFieldLength provided on IndexWriter. This patch also fixes a bug in the offset calculation in CharTokenizer (branch 3.x)  
  
Project: manifoldcf-integration-solr-3.x.txt  
Commit: ee586008a52b07a9a81e65c6fdbebfecc329704b  
ClassName: org.apache.lucene.analysis.CharTokenizer.java  
  
@@ -334,6 +341,8 @@ private boolean incrementTokenOld() throws IOException {  
- if (length == 0) // start of token  
+ if (length == 0) { // start of token  
**+ assert start == -1;** start = offset + bufferIndex - 1;  
- else if (length == buffer.length)  
+ } else if (length == buffer.length) {  
 buffer = termAtt.resizeBuffer(1+length);  
+ }  
========================================

LUCENE-2295: Added a LimitTokenCountAnalyzer / LimitTokenCountFilter to wrap any other Analyzer and provide the same functionality as MaxFieldLength provided on IndexWriter. This patch also fixes a bug in the offset calculation in CharTokenizer (branch 3.x)  
  
Project: manifoldcf-integration-solr-3.x.txt  
Commit: ee586008a52b07a9a81e65c6fdbebfecc329704b  
ClassName: org.apache.lucene.analysis.CharTokenizer.java  
  
@@ -287,6 +289,8 @@ public final boolean incrementToken() throws IOException {  
 if (isTokenChar(c)) { // if it's a token char  
- if (length == 0) // start of token  
+ if (length == 0) { // start of token  
**+ assert start == -1;** start = offset + bufferIndex - 1;  
- else if (length >= buffer.length-1) // check if a supplementary could run out of bounds  
+ } else if (length >= buffer.length-1) { // check if a supplementary could run out of bounds  
 buffer = termAtt.resizeBuffer(2+length); // make sure a supplementary fits in the buffer  
+ }  
 length += Character.toChars(normalize(c), buffer, length); // buffer it, normalized  
========================================

LUCENE-2593 (on 3x): fix certain rare cases where disk full could cause index corruption  
  
Project: manifoldcf-integration-solr-3.x.txt  
Commit: a27b2c434adbece0e04c1a91b1c6163f0097ad98  
ClassName: org.apache.lucene.index.IndexWriter.java  
  
 synchronized void close() throws IOException {  
**+ assert Thread.holdsLock(IndexWriter.this);** Iterator<Map.Entry<SegmentInfo,SegmentReader>> iter = readerMap.entrySet().iterator();  
========================================

LUCENE-2664: add SimpleText codec; fix -Dtests.seed to XX:YY format  
  
Project: manifoldcf-integration-solr-4.x.txt  
Commit: bac0a0435665bfdfc62d6c50b3ce849433ed6ca7  
ClassName: org.apache.lucene.index.codecs.simpletext.SimpleTextFieldsReader.java  
  
+ public BytesRef next() throws IOException {  
**+ assert !ended;**+ readLine(in, scratch);  
+ if (scratch.equals(END) || scratch.startsWith(FIELD)) {  
+ ended = true;  
+ current = null;  
+ return null;  
+ } else {  
**+ assert scratch.startsWith(TERM): "got " + scratch.utf8ToString();**+ docsStart = in.getFilePointer();  
+ final int len = scratch.length - TERM.length;  
+ if (len > scratch2.length) {  
+ scratch2.grow(len);  
+ }  
+ System.arraycopy(scratch.bytes, TERM.length, scratch2.bytes, 0, len);  
+ scratch2.length = len;  
+ current = scratch2;  
+ docFreq = 0;  
+ long lineStart = 0;  
+ while(true) {  
+ lineStart = in.getFilePointer();  
+ readLine(in, scratch);  
+ if (scratch.equals(END) || scratch.startsWith(FIELD) || scratch.startsWith(TERM)) {  
+ break;  
+ }  
+ if (scratch.startsWith(DOC)) {  
+ docFreq++;  
+ }  
+ }  
+ in.seek(lineStart);  
+ return current;  
+ }  
+ }  
========================================

**SOLR-2092: fix off-by-one in assert**  
Project: manifoldcf-integration-solr-4.x.txt  
Commit: a70430297a99d1de3e9f33d54a5150fbe133525e  
ClassName: org.apache.solr.request.UnInvertedField.java  
  
@@ -601,3 +601,3 @@ public NamedList getCounts(SolrIndexSearcher searcher, DocSet baseDocs, int offs  
 int collectCount = Math.max(0, queue.size() - off);  
**- assert collectCount < lim;  
+ assert collectCount <= lim;**========================================

LUCENE-2593: fix certain rare cases where disk full could cause index corruption  
  
Project: manifoldcf-integration-solr-4.x.txt  
Commit: e2f7336361b510df6b90acf9dae87935660dde63  
ClassName: org.apache.lucene.index.IndexWriter.java  
  
 synchronized void close() throws IOException {  
**+ assert Thread.holdsLock(IndexWriter.this);** Iterator<Map.Entry<SegmentInfo,SegmentReader>> iter = readerMap.entrySet().iterator();  
========================================

LUCENE-2664: add SimpleText codec; fix -Dtests.seed to XX:YY format  
  
Project: manifoldcf-integration-solr-4.x.txt  
Commit: bac0a0435665bfdfc62d6c50b3ce849433ed6ca7  
ClassName: org.apache.lucene.index.codecs.simpletext.SimpleTextFieldsReader.java  
  
+ public int nextPosition() throws IOException {  
+ readLine(in, scratch);  
**+ assert scratch.startsWith(POS): "got line=" + scratch.utf8ToString();**+ final int pos = Integer.parseInt(new String(scratch.bytes, scratch.offset+POS.length, scratch.length-POS.length));  
+ final long fp = in.getFilePointer();  
+ readLine(in, scratch);  
+ if (scratch.startsWith(PAYLOAD)) {  
+ final int len = scratch.length - PAYLOAD.length;  
+ if (scratch2.bytes.length < len) {  
+ scratch2.grow(len);  
+ }  
+ System.arraycopy(scratch.bytes, PAYLOAD.length, scratch2.bytes, 0, len);  
+ scratch2.length = len;  
+ payload = scratch2;  
+ } else {  
+ payload = null;  
+ in.seek(fp);  
+ }  
+ return pos;  
+ }  
========================================

LUCENE-2295: Added a LimitTokenCountAnalyzer / LimitTokenCountFilter to wrap any other Analyzer and provide the same functionality as MaxFieldLength provided on IndexWriter. This patch also fixes a bug in the offset calculation in CharTokenizer  
  
Project: manifoldcf-integration-solr-4.x.txt  
Commit: 7bb85177dfc690d25f35225d8946977f9389d71a  
ClassName: org.apache.lucene.analysis.CharTokenizer.java  
  
@@ -334,6 +341,8 @@ private boolean incrementTokenOld() throws IOException {  
- if (length == 0) // start of token  
+ if (length == 0) { // start of token  
**+ assert start == -1;** start = offset + bufferIndex - 1;  
- else if (length == buffer.length)  
+ } else if (length == buffer.length) {  
 buffer = termAtt.resizeBuffer(1+length);  
+ }  
========================================

SOLR-2089: fixed tests + exception when facet.offset is beyond queue size  
  
Project: manifoldcf-integration-solr-4.x.txt  
Commit: 6cfc766a34c0770c9646ced65062ebc5a6bb5aaf  
ClassName: org.apache.solr.request.UnInvertedField.java  
  
@@ -602,4 +602,6 @@ public NamedList getCounts(SolrIndexSearcher searcher, DocSet baseDocs, int offs  
- final int[] tnums = new int[Math.min(queue.size()-off, lim)];  
+ final int[] tnums = new int[Math.min(Math.max(0, queue.size()-off), lim)];  
 final int[] indirect = counts; // reuse the counts array for the index into the tnums array  
**+ assert indirect.length >= tnums.length;** int tnumCount = 0;  
========================================

LUCENE-2722: fix sep codec to store less in the terms dict  
  
Project: manifoldcf-integration-solr-4.x.txt  
Commit: 3d6111797eb4c35737f264df15c10292fc62635d  
ClassName: org.apache.lucene.index.codecs.intblock.FixedIntBlockIndexOutput.java  
  
+ public void write(IntIndexOutput indexOut, boolean absolute) throws IOException {  
+ if (absolute) {  
+ indexOut.writeVLong(fp);  
+ indexOut.write(upto);  
+ } else if (fp == lastFP) {  
+ indexOut.writeVLong(0);  
**+ assert upto >= lastUpto;**+ indexOut.write(upto - lastUpto);  
+ } else {   
+ indexOut.writeVLong(fp - lastFP);  
+ indexOut.write(upto);  
 }  
========================================

LUCENE-2722: fix sep codec to store less in the terms dict  
  
Project: manifoldcf-integration-solr-4.x.txt  
Commit: 3d6111797eb4c35737f264df15c10292fc62635d  
ClassName: org.apache.lucene.index.codecs.intblock.VariableIntBlockIndexOutput.java  
  
+ public void write(IntIndexOutput indexOut, boolean absolute) throws IOException {  
**+ assert upto >= 0;**+ if (absolute) {  
+ indexOut.writeVLong(fp);  
+ indexOut.write(upto);  
+ } else if (fp == lastFP) {  
+ indexOut.writeVLong(0);  
**+ assert upto >= lastUpto;**+ indexOut.write(upto);  
+ } else {   
+ indexOut.writeVLong(fp - lastFP);  
+ indexOut.write(upto);  
+ }  
+ lastUpto = upto;  
+ lastFP = fp;  
+ }  
========================================

LUCENE-2593: fix certain rare cases where disk full could cause index corruption  
  
Project: manifoldcf-integration-solr-4.x.txt  
Commit: e2f7336361b510df6b90acf9dae87935660dde63  
ClassName: org.apache.lucene.index.IndexWriter.java  
  
 synchronized void commit() throws IOException {  
**+ assert Thread.holdsLock(IndexWriter.this);** for (Map.Entry<SegmentInfo,SegmentReader> ent : readerMap.entrySet()) {  
========================================

SOLR-1297: fix sort by function parsing  
  
Project: manifoldcf-integration-solr-4.x.txt  
Commit: 62d8c5cb9a49ca86e549e7fcb0b2dca8fc325778  
ClassName: org.apache.solr.search.QueryParsing.java  
  
- public static Sort parseSort(String sortSpec, IndexSchema schema) {  
+ public static Sort parseSort(String sortSpec, SolrQueryRequest req) {  
 if (sortSpec == null || sortSpec.length() == 0) return null;  
- char[] chars = sortSpec.toCharArray();  
- int i = 0;  
- StringBuilder buffer = new StringBuilder(sortSpec.length());  
- String sort = null;  
- String order = null;  
- int functionDepth = 0;  
- boolean score = true;  
- List<SortField> lst = new ArrayList<SortField>(5);  
- boolean needOrder = false;  
- while (i < chars.length) {  
- if (Character.isWhitespace(chars[i]) && functionDepth == 0) {  
- if (buffer.length() == 0) {  
- } else {  
- if (needOrder == false) {  
- sort = buffer.toString().trim();  
- buffer.setLength(0);  
- needOrder = true;  
- } else {  
- order = buffer.toString().trim();  
- buffer.setLength(0);  
- needOrder = false;  
- }  
- }  
- } else if (chars[i] == '(' && functionDepth >= 0) {  
- buffer.append(chars[i]);  
- functionDepth++;  
- } else if (chars[i] == ')' && functionDepth > 0) {  
- buffer.append(chars[i]);  
- functionDepth--;//close up one layer  
- } else if (chars[i] == ',' && functionDepth == 0) {//can either be a separator of sort declarations, or a separator in a function  
- if (needOrder == true && buffer.length() > 0){  
- order = buffer.toString().trim();  
- buffer.setLength(0);  
- needOrder = false;  
- }  
- score = processSort(schema, sort, order, lst);  
- sort = null;  
- order = null;  
- buffer.setLength(0);//get ready for the next one, if there is one  
- } else if (chars[i] == ',' && functionDepth > 0) {  
- buffer.append(chars[i]);  
+ List<SortField> lst = new ArrayList<SortField>(4);  
+ try {  
+ StrParser sp = new StrParser(sortSpec);  
+ while (sp.pos < sp.end) {  
+ sp.eatws();  
+ int start = sp.pos;  
+ String field = sp.getId(null);  
+ ValueSource vs = null;  
+ if (field == null || sp.ch() != ' ') {  
+ String funcStr = sp.val.substring(start);  
+ QParser parser = QParser.getParser(funcStr, FunctionQParserPlugin.NAME, req);  
+ Query q = null;  
+ if (parser instanceof FunctionQParser) {  
+ FunctionQParser fparser = (FunctionQParser)parser;  
+ fparser.setParseMultipleSources(false);  
+ fparser.setParseToEnd(false);  
+ q = fparser.getQuery();  
+ if (fparser.localParams != null) {  
+ if (fparser.valFollowedParams) {  
+ int leftOver = fparser.sp.end - fparser.sp.pos;  
+ sp.pos = sp.end - leftOver; // reset our parser to the same amount of leftover  
 } else {  
- buffer.append(chars[i]);  
+ sp.pos = start + fparser.localParamsEnd;  
 }  
- i++;  
- }  
- if (buffer.length() > 0 && needOrder){//see if we have anything left, at most it should be an order  
- order = buffer.toString().trim();  
- buffer.setLength(0);  
- needOrder = false;  
+ } else {  
+ int leftOver = fparser.sp.end - fparser.sp.pos;  
+ sp.pos = sp.end - leftOver; // reset our parser to the same amount of leftover  
 }  
+ } else {  
+ q = parser.getQuery();  
- if (functionDepth != 0){  
- throw new SolrException(SolrException.ErrorCode.BAD\_REQUEST, "Unable to parse sort spec, mismatched parentheses: " + sortSpec);  
- }  
- if (buffer.length() > 0){//there's something wrong, as everything should have been parsed by now  
- throw new SolrException(SolrException.ErrorCode.BAD\_REQUEST, "Unable to parse sort spec: " + sortSpec);  
**+ assert parser.getLocalParams() != null;**+ sp.pos = start + parser.localParamsEnd;  
 }  
  
- score = processSort(schema, sort, order, lst);  
- }  
- if (lst.size() == 1 && score == true && lst.get(0).getReverse() == false) {  
- return null; // do normal scoring...  
+ if (q instanceof FunctionQuery) {  
+ vs = ((FunctionQuery)q).getValueSource();  
+ } else {  
+ vs = new QueryValueSource(q, 0.0f);  
 }  
- return new Sort((SortField[]) lst.toArray(new SortField[lst.size()]));  
 }  
- private static boolean processSort(IndexSchema schema, String sort, String order, List<SortField> lst) {  
- boolean score = false;  
- if (sort != null && order != null) {  
- boolean top = true;  
+ String order = sp.getId("Expected sort order asc/desc");  
+ boolean top;  
 if ("desc".equals(order) || "top".equals(order)) {  
========================================

Resolving issue: MNG-522  
o Added check for null child pluginManagement section, and setting to parent value if null.  
o Added integration test for parent->child pluginManagement inheritance and injection, to prevent regressions.  
  
Project: maven.txt  
Commit: e4083dcdea2f468569f76943de2c2e1276784a70  
ClassName: org.apache.maven.it0001.Person.java  
  
+ public void setName( String newName )  
+ {  
**+ assert true;**+ this.name = newName;  
+ }  
========================================

MNG-4916 cache extension plugin resolution errors  
  
Project: maven.txt  
Commit: e2600769242dcbee9702a40fe6754f767fe2dd8d  
ClassName: org.apache.maven.plugin.DefaultPluginArtifactsCache.java  
  
@@ -155,2 +163,14 @@ public CacheRecord put( Key key, List<Artifact> pluginArtifacts )  
**+ assertUniqueKey( key );**+ CacheRecord record =  
+ new CacheRecord( Collections.unmodifiableList( new ArrayList<Artifact>( pluginArtifacts ) ) );  
+ cache.put( key, record );  
+ return record;  
+ }  
**+ protected void assertUniqueKey( Key key )**+ {  
========================================

[DOXIASITETOOLS-24] SiteRendererSink swallows formatting in section titles. Also fix anchor generation for section titles.  
  
Project: maven-doxia-sitetools.txt  
Commit: ffdb7aa84e1019f419a8360f06d8c85e2e60aa1c  
ClassName: org.apache.maven.doxia.siterenderer.NestedItemsVerifier.java  
  
@@ -260,2 +268,6 @@ public void verify( String file )  
+ a = (HtmlAnchor) elementIterator.next();  
**+ assertNotNull( a );  
+ assertEquals( a.getAttributeValue( "name" ), "Definition\_lists" );** p = (HtmlParagraph) elementIterator.next();  
========================================

[DOXIASITETOOLS-24] SiteRendererSink swallows formatting in section titles. Also fix anchor generation for section titles.  
  
Project: maven-doxia-sitetools.txt  
Commit: ffdb7aa84e1019f419a8360f06d8c85e2e60aa1c  
ClassName: org.apache.maven.doxia.siterenderer.EntitiesVerifier.java  
  
@@ -169,2 +173,6 @@ public void verify( String file )  
+ a = (HtmlAnchor) elementIterator.next();  
**+ assertNotNull( a );  
+ assertEquals( a.getAttributeValue( "name" ), "CDATA" );** div = (HtmlDivision) elementIterator.next();  
========================================

Another thing to fix: default entities like ampersand get un-escaped in head. See Doxia-309.  
  
Project: maven-doxia-sitetools.txt  
Commit: 70d9f3fb3f99bfda506a10d736d1b1a727251913  
ClassName: org.apache.maven.doxia.siterenderer.EntitiesVerifier.java  
  
@@ -54,2 +54,7 @@ public void verify( String file )  
+ author = (HtmlMeta) page.getHtmlElementsByName( "author" ).get( 1 );  
**+ assertNotNull( author );** HtmlElement element = page.getHtmlElementById( "contentBox" );  
========================================

More tests to fix for DOXIA-309, DOXIA-310, DOXIA-311.  
  
Project: maven-doxia-sitetools.txt  
Commit: fa9dc3a369f7db0c15964ba8d3610ee795191c8e  
ClassName: org.apache.maven.doxia.siterenderer.EntitiesVerifier.java  
  
@@ -88,3 +90,3 @@ public void verify( String file )  
 **assertNotNull( h3 );  
- assertEquals( h3.asText().trim(), "Generic Entities" );  
+ assertEquals( h3.asText().trim(), "Generic Entities: '&' '<' '>' '\"' '''" );**========================================

[DOXIASITETOOLS-24] SiteRendererSink swallows formatting in section titles. Also fix anchor generation for section titles.  
  
Project: maven-doxia-sitetools.txt  
Commit: ffdb7aa84e1019f419a8360f06d8c85e2e60aa1c  
ClassName: org.apache.maven.doxia.siterenderer.FaqVerifier.java  
  
@@ -71,2 +71,5 @@ public void verify( String file )  
 HtmlAnchor a = (HtmlAnchor) elementIterator.next();  
**+ assertEquals( a.getAttributeValue( "name" ), "top" );**+ a = (HtmlAnchor) elementIterator.next();  
 **assertEquals( a.getAttributeValue( "name" ), "Oft\_Asked\_Questions" );**========================================

[DOXIASITETOOLS-24] SiteRendererSink swallows formatting in section titles. Also fix anchor generation for section titles.  
  
Project: maven-doxia-sitetools.txt  
Commit: ffdb7aa84e1019f419a8360f06d8c85e2e60aa1c  
ClassName: org.apache.maven.doxia.siterenderer.AptVerifier.java  
  
@@ -100,2 +102,61 @@ public void verify( String file )  
 **assertEquals( "/index.html", a.getAttributeValue( "href" ) );**+ div = (HtmlDivision) elementIterator.next();  
**+ assertEquals( "section", div.getAttributeValue( "class" ) );**+ h2 = (HtmlHeader2) elementIterator.next();  
**+ assertNotNull( h2 );  
+ assertEquals( "Section formatting: italic bold mono", h2.asText().trim() );**+ UnknownHtmlElement unknown = (UnknownHtmlElement) elementIterator.next();  
**+ assertEquals( "i", unknown.getTagName() );  
+ assertEquals( "italic", unknown.asText().trim() );**+ unknown = (UnknownHtmlElement) elementIterator.next();  
**+ assertEquals( "b", unknown.getTagName() );  
+ assertEquals( "bold", unknown.asText().trim() );**+ unknown = (UnknownHtmlElement) elementIterator.next();  
**+ assertEquals( "tt", unknown.getTagName() );  
+ assertEquals( "mono", unknown.asText().trim() );**+ a = (HtmlAnchor) elementIterator.next();  
**+ assertEquals( "Section\_formatting:\_italic\_bold\_mono", a.getAttributeValue( "name" ) );**+ div = (HtmlDivision) elementIterator.next();  
**+ assertEquals( "section", div.getAttributeValue( "class" ) );**+ HtmlHeader3 h3 = (HtmlHeader3) elementIterator.next();  
**+ assertNotNull( h3 );  
+ assertEquals( "SubSection formatting: italic bold mono", h3.asText().trim() );**+ unknown = (UnknownHtmlElement) elementIterator.next();  
**+ assertEquals( "i", unknown.getTagName() );  
+ assertEquals( "italic", unknown.asText().trim() );**+ unknown = (UnknownHtmlElement) elementIterator.next();  
**+ assertEquals( "b", unknown.getTagName() );  
+ assertEquals( "bold", unknown.asText().trim() );**+ unknown = (UnknownHtmlElement) elementIterator.next();  
**+ assertEquals( "tt", unknown.getTagName() );  
+ assertEquals( "mono", unknown.asText().trim() );**+ a = (HtmlAnchor) elementIterator.next();  
**+ assertEquals( "SubSection\_formatting:\_italic\_bold\_mono", a.getAttributeValue( "name" ) );**+ p = (HtmlParagraph) elementIterator.next();  
**+ assertNotNull( p );**+ unknown = (UnknownHtmlElement) elementIterator.next();  
**+ assertEquals( "i", unknown.getTagName() );  
+ assertEquals( "italic", unknown.asText().trim() );**+ unknown = (UnknownHtmlElement) elementIterator.next();  
**+ assertEquals( "b", unknown.getTagName() );  
+ assertEquals( "bold", unknown.asText().trim() );**+ unknown = (UnknownHtmlElement) elementIterator.next();  
**+ assertEquals( "tt", unknown.getTagName() );  
+ assertEquals( "mono", unknown.asText().trim() );** }  
========================================

o fix warning  
  
Project: maven-doxia-sitetools.txt  
Commit: 976c771f84748f066ddfb1d23d58e35823ad8c2d  
ClassName: org.apache.maven.doxia.siterenderer.AptVerifier.java  
  
@@ -98,3 +98,3 @@ public void verify( String file )  
 a = (HtmlAnchor) elementIterator.next();  
**- assertEquals( "#cdc.html", a.getAttributeValue( "href" ) );  
+ assertEquals( "#cdc.html.internal", a.getAttributeValue( "href" ) );**========================================

[MECLIPSE-486] fixing ITs to support multi-module expected file checking.  
pom.xml:  
Fixed the phase "test" to not include "projects/\*\*" as this directory is copied from src/test/resources into target/test-classes  
and surefire will attempt to compile any \*Test\*.java files - which will fail since they are not in the correct location  
relative to their package declaration. See http://www.nabble.com/Plugins%3A-execution-configuration-does-not-add-to-configuration-td19425025.html  
Fixed the run-its profile so that it overrides the excludes inherited from the "test" phase configuration.  
IdeUtils:  
Refactored code to create fixSeparator() method.  
AbstractEclipsePluginIT:  
Changed compareDirectoryContent to no longer accept a directory to use as check contents.  
Instead ${basedir} is searched for EXPECTED\_DIRECTORY\_NAME ("expected") and all occurrences  
of this directory are used in the comparison processes. This has simplified checking in  
code as you no longer need to specify additional calls for multiple directories and  
made it easier to support multi-module checking as previously only ${basedir}/expected  
was checked, module directories where ignored. The actual files checked are relative to the  
parent directory of the matching "expected" directory.  
IOExceptions are no longer thrown instead MojoExecutionException are used mainly by  
delegating to IdeUtils.getCanonicalPath() instead of calling File.getCanonicalPath() directly.  
RadPluginIT:  
Updated to use new compareDirectoryContent()  
src/test/resources/projects/\*:  
Fixed files that were never previously checked to use the contents of the actual  
maven-eclipse-plugin output. On the assumption that the plugin is generating the correct  
output and nothing in this change set has altered that behaviour.  
  
Project: maven-plugins.txt  
Commit: 7b5b89e0986a99b6d52e9867aabf6f1b7755c317  
ClassName: org.apache.maven.plugin.eclipse.it.AbstractEclipsePluginIT.java  
  
**@@ -532,3 +525,4 @@ protected void assertFileEquals( File expectedFile, File actualFile )  
- assertEquals( "Checking " + filename + ", line #" + ( i + 1 ), expected, actual );  
+ assertEquals( "Comparing '" + IdeUtils.getCanonicalPath( actualFile ) + "' against '"**+ + IdeUtils.getCanonicalPath( expectedFile ) + "' at line #" + ( i + 1 ), expected, actual );  
 }  
========================================

fix some issues in the dependency report (wrong dependency versions included), and simplify  
  
Project: maven-plugins.txt  
Commit: ad45fa36263c297386288d390ebf3f7d58bbbbdd  
ClassName: org.apache.maven.report.projectinfo.ReportResolutionListener.java  
  
- private void addToDepTree( Artifact artifact )  
- {  
- if ( parents.size() > 0 )  
+ public void omitForNearer( Artifact omitted, Artifact kept )  
 {  
- Artifact parent = (Artifact) parents.get( parents.size() - 1 );  
**+ assert omitted.getDependencyConflictId().equals( kept.getDependencyConflictId() );**- if ( depTree.containsKey( parent.getId() ) )  
+ Node prev = (Node) artifacts.get( omitted.getDependencyConflictId() );  
+ if ( prev != null )  
 {  
- List deps = (List) depTree.get( parent.getId() );  
- deps.add( artifact );  
- }  
- else  
+ if ( prev.parent != null )  
 {  
- List deps = new ArrayList();  
- deps.add( artifact );  
- depTree.put( parent.getId(), deps );  
- }  
+ prev.parent.children.remove( prev );  
 }  
+ artifacts.remove( omitted.getDependencyConflictId() );  
 }  
========================================

[MSUREFIRE-86] fix remaining issues with forking on 1.3  
  
Project: maven-surefire.txt  
Commit: c98d44e8b1f22aa783d9fa290ede5097b6874f71  
ClassName: org.apache.maven.surefire.booter.SurefireBooter.java  
  
@@ -534,6 +535,18 @@ private static ClassLoader createClassLoader( List classPathUrls, ClassLoader pa  
 IsolatedClassLoader classLoader = new IsolatedClassLoader( parent, childDelegation );  
**- if ( assertionsAvailable )  
+ if ( assertionStatusMethod != null )** {  
**- parent.setDefaultAssertionStatus( assertionsEnabled );  
- classLoader.setDefaultAssertionStatus( assertionsEnabled );**+ try  
+ {  
**+ Object[] args = new Object[]{assertionsEnabled ? Boolean.TRUE : Boolean.FALSE};  
+ assertionStatusMethod.invoke( parent, args );  
+ assertionStatusMethod.invoke( classLoader, args );**+ }  
+ catch ( IllegalAccessException e )  
+ {  
**+ throw new NestedRuntimeException( "Unable to access the assertion enablement method", e );**+ }  
+ catch ( InvocationTargetException e )  
+ {  
**+ throw new NestedRuntimeException( "Unable to invoke the assertion enablement method", e );**+ }  
 }  
========================================

[SUREFIRE-584] Fix the verifier based tests so that they work during mvn release:prepare  
o I have used invoker:install to stage the artifacts required for the integration tests  
o I use a JUnit test to setup the settings.xml that uses the staged local repository and uses the actual local repository as a remote repository while retaining all the running machine's settings (e.g. proxies) that should mean that the integration tests will work from behind a corporate proxy and they should be fast  
o I use both surefire and failsafe maven plugins for test execution, failsafe is used for the integration tests and surefire runs the JUnit test that creates the settings.xml for the staged local repository. I could have just used exec-maven-plugin and used a executible java class to create the staged settings.xml, but my concern is that that would introduce another dependency and complicate bootstrapping... failsafe will not be an issue in this respect as I will be adding it to surefire.  
o All the integration tests have been renamed to be picked up by failsafe's default patterns.  
  
Project: maven-surefire.txt  
Commit: af8367610a0fd702c874c08637b48cf8c2ce60fa  
ClassName: org.apache.maven.surefire.its.ForkModeIT.java  
  
**+ private void assertDifferentPids( String[] pids )**+ {  
+ if ( pids[0].equals( pids[1] ) )  
+ {  
+ fail( "pid 1 matched pid 2: " + pids[0] );  
+ }  
+ if ( pids[0].equals( pids[2] ) )  
+ {  
+ fail( "pid 1 matched pid 3: " + pids[0] );  
+ }  
+ if ( pids[1].equals( pids[2] ) )  
+ {  
+ fail( "pid 2 matched pid 3: " + pids[0] );  
+ }  
+ }  
========================================

o Split surefire booter in 2. Refactoring only, no functional change.  
Extracted BooterConfiguration from the booter, since the booter class was  
just way too large. BooterConfiguration should now be easily unit-testable too,  
which is good because there are a fair number of unsolved issues there.  
There are still some smaller refactorings that can be done, including fine-tuning  
of some of the responsibilities between SurefireBooter, BooterConfiguration and  
possibly one or two more helper classes. But this commit represents the  
heavy-lifting work of splitting this huge class into pieces.  
  
Project: maven-surefire.txt  
Commit: ff212930017bc7875f1c597379f70fdce59da4e3  
ClassName: org.apache.maven.surefire.booter.SurefireBooter.java  
  
@@ -753,3 +519,3 @@ private ClassLoader createClassLoader( List classPathUrls, ClassLoader parent, b  
 {  
**- Object[] args = new Object[] { enableAssertions ? Boolean.TRUE : Boolean.FALSE };  
+ Object[] args = new Object[]{ booterConfiguration.isEnableAssertions() ? Boolean.TRUE : Boolean.FALSE };** if ( parent != null )  
========================================

Revert "[SUREFIRE-700] Isolate surefire from itself"  
It broke the windows build and I won't have time to  
fix it before going off on more vacation. Someone's  
got to try to beat Julia in # vacation days ;)  
  
Project: maven-surefire.txt  
Commit: 61d4be622eeb43e1ca479846c721041f64d4b73c  
ClassName: org.apache.maven.surefire.booter.BooterDeserializer.java  
  
@@ -111,3 +110,3 @@ public StartupConfiguration getProviderConfiguration()  
 ClasspathConfiguration classpathConfiguration =  
**- new ClasspathConfiguration( classpath, sureFireClasspath, testFrameworkClasspath, enableAssertions, childDelegation );  
+ new ClasspathConfiguration( classpath, sureFireClasspath, enableAssertions, childDelegation );**========================================

[SUREFIRE-321] Added runOrder mojo attribute, including documentation.  
Also added IT for the various runOrder settings.  
Also fixed minor regression in selection of reporters  
  
Project: maven-surefire.txt  
Commit: 5260100a75ac9d2605994e22e0ef29bdb9666730  
ClassName: org.apache.maven.surefire.its.RunOrderIT.java  
  
+ private List executeWithRunOrder( String runOrder )  
+ throws IOException, VerificationException, MavenReportException  
+ {  
+ File testDir = ResourceExtractor.simpleExtractResources( getClass(), "/runOrder" );  
+ Verifier verifier = new Verifier( testDir.getAbsolutePath() );  
+ List goals = getInitialGoals();  
+ goals.add( "-DrunOrder=" + runOrder );  
+ goals.add( "test" );  
+ this.executeGoals( verifier, goals );  
+ verifier.verifyErrorFreeLog();  
+ verifier.resetStreams();  
**+ HelperAssertions.assertTestSuiteResults( 3, 0, 0, 0, testDir );**+ return verifier.loadFile( verifier.getBasedir(), verifier.getLogFileName(), false );  
+ }  
========================================

[SUREFIRE-703] junit47 does not work with redirectToTestOutputToFile  
Integration test submitted by Will May. The issue itself was fixed in SUREFIRE-732 refactoring  
  
Project: maven-surefire.txt  
Commit: 8d8e86f684b11d8976da2eea29b343142a3002b2  
ClassName: org.apache.maven.surefire.its.JUnit47RedirectOutputIT.java  
  
+ private void checkReports()  
+ throws IOException  
+ {  
+ String report = StringUtils.trimToNull(  
+ FileUtils.readFileToString( getSurefireReportsFile( "junit47ConsoleOutput.Test1-output.txt" ) ) );  
**+ assertNotNull( report );**+ String report2 = StringUtils.trimToNull( FileUtils.readFileToString(  
+ getSurefireReportsFile( "junit47ConsoleOutput.Test2-output.txt" ) ) );  
**+ assertNotNull(report2);**+ }  
========================================

**[SUREFIRE-35] fix assertion enablement in the 4 available class loading scenarios**  
Project: maven-surefire.txt  
Commit: 8bdaf01d68e029f52eac12caa31a2f5669cc8c2d  
ClassName: org.apache.maven.surefire.booter.SurefireBooter.java  
  
@@ -519,6 +534,7 @@ private static ClassLoader createClassLoader( List classPathUrls, ClassLoader pa  
 IsolatedClassLoader classLoader = new IsolatedClassLoader( parent, childDelegation );  
**+ if ( assertionsAvailable )**+ {  
**+ parent.setDefaultAssertionStatus( assertionsEnabled );  
+ classLoader.setDefaultAssertionStatus( assertionsEnabled );**+ }  
========================================

o Split surefire booter in 2. Refactoring only, no functional change.  
Extracted BooterConfiguration from the booter, since the booter class was  
just way too large. BooterConfiguration should now be easily unit-testable too,  
which is good because there are a fair number of unsolved issues there.  
There are still some smaller refactorings that can be done, including fine-tuning  
of some of the responsibilities between SurefireBooter, BooterConfiguration and  
possibly one or two more helper classes. But this commit represents the  
heavy-lifting work of splitting this huge class into pieces.  
  
Project: maven-surefire.txt  
Commit: ff212930017bc7875f1c597379f70fdce59da4e3  
ClassName: org.apache.maven.surefire.booter.BooterConfiguration.java  
  
**+ public boolean isEnableAssertions()**+ {  
**+ return enableAssertions;**+ }  
========================================

Resolved issue: DIRMINA-440 (Consistent naming style for acronyms (SSL, HTTP, MDC, IO, ...))  
  
Project: mina.txt  
Commit: cac05068ea007b36449c9a67d39fb446c837a145  
ClassName: org.apache.mina.integration.spring.ssl.SslContextFactoryBean.java  
  
+ public void setProtocol(String protocol) {  
**+ Assert.notNull(protocol, "Property 'protocol' may not be null");**+ this.protocol = protocol;  
+ }  
========================================

Fixed compilation errors.  
  
Project: mina.txt  
Commit: 7983add50a66e1e8d610130dafe74732bec7043e  
ClassName: org.apache.mina.integration.spring.support.AbstractIoServiceFactoryBean.java  
  
+ public void setFilterMappings( IoFilterMapping[] filterMappings )  
+ {  
**+ Assert.notNull( filterMappings, "Property 'filterMappings' may not be null" );**+ this.filterMappings = filterMappings;  
+ }  
========================================

Resolved issue: DIRMINA-378 (Reformat code using new coding conventions)  
\* Reformatted all code using Eclipse formatter  
  
Project: mina.txt  
Commit: c6998fd4c565af8cf0ef516be0428ad5f15b5cde  
ClassName: org.apache.mina.common.support.AbstractIoFilterChain.java  
  
 public void filterWrite(NextFilter nextFilter, IoSession session,  
- WriteRequest writeRequest ) throws Exception  
- {  
+ WriteRequest writeRequest) throws Exception {  
**- assert session.getTransportType().getEnvelopeType().isAssignableFrom( writeRequest.getMessage().getClass() );  
+ assert session.getTransportType().getEnvelopeType()**+ .isAssignableFrom(writeRequest.getMessage().getClass());  
 doWrite(session, writeRequest);  
========================================

https://issues.apache.org/jira/browse/MYFACES-2039  
  
Project: myfaces.txt  
Commit: 48f6a26aaa68968b81c462b19b1a6d8473fe3c77  
ClassName: org.apache.myfaces.context.servlet.FacesContextImpl.java  
  
 public boolean isPostback() {  
**+ assertNotReleased(METHOD\_RENDERRESPONSE);** return getRenderKit().getResponseStateManager().isPostback(this);  
========================================

https://issues.apache.org/jira/browse/MYFACES-2118  
added the render view code for the ppr response!  
  
Project: myfaces.txt  
Commit: 376d34160e4b90390d45aacad06d48362750c296  
ClassName: org.apache.myfaces.context.servlet.PartialViewContextImpl.java  
  
@@ -265,6 +294,4 @@ private void processPartialRendering(UIComponent viewRoot, PhaseId phaseId) {  
**- private void assertNotReleased()**- {  
- if (\_released)  
- {  
**+ private void assertNotReleased() {**+ if (\_released) {  
 throw new IllegalStateException("Error the FacesContext is already released!");  
========================================

MYFACES-2334 get ifnavigation sample working (fix toViewId when use ValueExpression)  
  
Project: myfaces.txt  
Commit: ca8e13ccc455b46a69bbf2907981f6cbe23d85d5  
ClassName: javax.faces.application.NavigationCase.java  
  
@@ -254,2 +263,15 @@ private ValueExpression \_getConditionExpression(FacesContext context)  
+ private ValueExpression \_getToViewIdExpression(FacesContext context)  
+ {  
**+ assert \_toViewId != null;**+ if (\_toViewIdExpression == null)  
+ {  
+ ExpressionFactory factory = context.getApplication().getExpressionFactory();  
+ \_toViewIdExpression = factory.createValueExpression(context.getELContext(), \_toViewId, String.class);  
+ }  
+ return \_toViewIdExpression;  
+ }  
========================================

https://issues.apache.org/jira/browse/MYFACES-2004  
  
Project: myfaces.txt  
Commit: 5913e1518401d08d9ec13c19f210a57ad4882f22  
ClassName: org.apache.myfaces.context.servlet.FacesContextImpl.java  
  
 public void setExecutePhaseClientIds(List<String> executePhaseClientIds) {  
**+ assertContextState("setExecutePhaseClientIds");** super.setExecutePhaseClientIds(executePhaseClientIds);  
========================================

MYFACES-1974 Implementing Java 5 features on myfaces-impl, first loop;  
\*\*\* IMPORTANT \*\*\* This patch might break backward compatibility for some people if extending impl directly.  
However, most of the time, it shouldn't be an issue as you're not supposed to extend impl in order to remain implementation agnostic.  
The main compatibility breaking change involve method signature alteration to make them return a Collection rather than an Iterator to leverage enhanced for looping.  
  
Project: myfaces.txt  
Commit: 4055108964b381a9e656e095cc4c0690c1387c1d  
ClassName: org.apache.myfaces.util.DebugUtils.java  
  
**@@ -92,4 +93,3 @@ public static void assertError(boolean condition, Log log\_, String msg)  
- public static void assertFatal(boolean condition, Log log\_, String msg)**- throws FacesException  
**+ public static void assertFatal(boolean condition, Log log\_, String msg) throws FacesException** {  
========================================

https://issues.apache.org/jira/browse/MYFACES-2241  
added the needed implementations for the vist context, next step is to enable it in our codebase  
  
Project: myfaces.txt  
Commit: 09fe24b362b74d98b3625adce82636e26c0b300b  
ClassName: org.apache.myfaces.component.visit.PartialVisitContext.java  
  
**+ private void assertNamingContainer(UIComponent component) throws IllegalArgumentException {**+ if (!(component instanceof NamingContainer)) {  
+ throw new IllegalArgumentException("Component " + component.getClientId(\_facesContext) + "must be of type NamingContainer");  
+ }  
+ }  
========================================

https://issues.apache.org/jira/browse/MYFACES-2004  
  
Project: myfaces.txt  
Commit: 5913e1518401d08d9ec13c19f210a57ad4882f22  
ClassName: org.apache.myfaces.context.servlet.FacesContextImpl.java  
  
 public void setRenderPhaseClientIds(List<String> renderPhaseClientIds) {  
- super.setRenderPhaseClientIds(renderPhaseClientIds);  
**+ assertContextState("setExecutePhaseClientIds");**+ \_renderPhaseClientIds = renderPhaseClientIds;  
 }  
========================================

https://issues.apache.org/jira/browse/MYFACES-1992  
  
Project: myfaces.txt  
Commit: 81fb18fa2a55cd51edf694236b97f6c03f8b9b41  
ClassName: org.apache.myfaces.context.servlet.FacesContextImpl.java  
  
 public boolean isAjaxRequest() {  
- Map requestMap = getExternalContext().getRequestMap();  
- return requestMap.containsKey(AJAX\_REQ\_KEY);  
- }\*/  
**+ assertNotReleased(METHOD\_ISAJAXREQUEST);**+ if (\_ajaxRequest == null) {  
========================================

https://issues.apache.org/jira/browse/MYFACES-2004  
  
Project: myfaces.txt  
Commit: 5913e1518401d08d9ec13c19f210a57ad4882f22  
ClassName: org.apache.myfaces.context.servlet.FacesContextImpl.java  
  
 public void enableResponseWriting(boolean enable) {  
**+ assertContextState("enableResponseWriting");** \_responseWrapper.setEnabled(enable);  
========================================

https://issues.apache.org/jira/browse/MYFACES-2039  
  
Project: myfaces.txt  
Commit: 48f6a26aaa68968b81c462b19b1a6d8473fe3c77  
ClassName: org.apache.myfaces.context.servlet.FacesContextImpl.java  
  
 public PhaseId getCurrentPhaseId() {  
**+ assertNotReleased(METHOD\_GETCURRENTPHASEID);** return \_currentPhaseId;  
========================================

https://issues.apache.org/jira/browse/EXTSCRIPT-98  
project reorg due to pending issues with the svn change history plugin  
  
Project: myfaces-scripting.txt  
Commit: e2504fe55933883525e442f49c54c3deae365b14  
ClassName: org.apache.myfaces.scripting.core.util.FileUtils.java  
  
**+ public static void assertPath(File path) {**+ if (!path.exists()) {  
+ if (!path.mkdirs()) {  
+ throw new IllegalStateException("It wasn't possible to create the target " +  
+ "directory for the compiler ['" + path.getAbsolutePath() + "'].");  
+ }  
+ path.deleteOnExit();  
+ }  
+ }  
========================================

https://issues.apache.org/jira/browse/EXTSCRIPT-3  
Major core refactoring which will make it easier in the near future to  
add new engines, the common non engine related functionality  
of the script weaving has been moved into a common base class  
  
Project: myfaces-scripting.txt  
Commit: d83711f325a3b77302707f30fb46317a10ce2147  
ClassName: org.apache.myfaces.scripting.core.BaseWeaver.java  
  
+ protected boolean isReloadCandidate(ReloadingMetadata reloadMeta) {  
**+ return reloadMeta != null && assertScriptingEngine(reloadMeta) && reloadMeta.isTaintedOnce();**+ }  
========================================

https://issues.apache.org/jira/browse/EXTSCRIPT-47  
refactoring for the facelet patchup part  
  
Project: myfaces-scripting.txt  
Commit: d754ac9b80642325f0ff0faeb2c82b761b6cab8d  
ClassName: org.apache.myfaces.scripting.facelet.support.SwitchingMetarulesetImpl.java  
  
+ public Metadata finish() {  
**+ assert !\_rules.isEmpty();**+ if (!\_attributes.isEmpty()) {  
+ MetadataTarget target = this.\_getMetadataTarget();  
+ int ruleEnd = \_rules.size() - 1;  
+ for (Map.Entry<String, TagAttribute> entry : \_attributes.entrySet()) {  
+ Metadata data = null;  
+ int i = ruleEnd;  
+ do {  
+ MetaRule rule = \_rules.get(i);  
+ data = rule.applyRule(entry.getKey(), entry.getValue(), target);  
+ i--;  
+ } while (data == null && i >= 0);  
+ if (data == null) {  
+ if (log.isLoggable(Level.SEVERE)) {  
+ log.severe(entry.getValue() + " Unhandled by MetaTagHandler for type " + \_type.getName());  
+ }  
+ } else {  
+ \_mappers.add(data);  
+ }  
+ }  
+ }  
+ if (\_mappers.isEmpty()) {  
+ return NONE;  
+ } else {  
+ return new MetadataImpl(\_mappers.toArray(new Metadata[\_mappers.size()]));  
+ }  
+ }  
========================================

https://issues.apache.org/jira/browse/EXTSCRIPT-3  
Major core refactoring which will make it easier in the near future to  
add new engines, the common non engine related functionality  
of the script weaving has been moved into a common base class  
  
Project: myfaces-scripting.txt  
Commit: d83711f325a3b77302707f30fb46317a10ce2147  
ClassName: org.apache.myfaces.scripting.core.BaseWeaver.java  
  
+ public Class reloadScriptingClass(Class aclass) {  
+ ReloadingMetadata metadata = getClassMap().get(aclass.getName());  
+ if (metadata == null)  
+ return aclass;  
**+ if(!assertScriptingEngine(metadata)) {**+ return null;  
+ }  
+ if (!metadata.isTainted()) {  
+ return metadata.getAClass();  
+ }  
+ return loadScriptingClassFromFile(metadata.getSourcePath(), metadata.getFileName());  
+ }  
========================================

https://issues.apache.org/jira/browse/EXTSCRIPT-98  
project reorg due to pending issues with the svn change history plugin  
  
Project: myfaces-scripting.txt  
Commit: 1aaefa3917af7a70f7127bce8786af7926ec85b1  
ClassName: org.apache.myfaces.scripting.api.BaseWeaver.java  
  
**+ protected boolean assertScriptingEngine(ReloadingMetadata reloadMeta) {**+ return reloadMeta.getScriptingEngine() == getScriptingEngine();  
+ }  
========================================

https://issues.apache.org/jira/browse/EXTSCRIPT-98  
project reorg due to pending issues with the svn change history plugin  
  
Project: myfaces-scripting.txt  
Commit: 1aaefa3917af7a70f7127bce8786af7926ec85b1  
ClassName: org.apache.myfaces.scripting.core.classIdentifier.DynamicClassloader.java  
  
+ public Class loadClass(String className, boolean resolve) throws ClassNotFoundException {  
+ if (className.contains("java.lang")) {  
+ return super.loadClass(className, resolve);  
+ }  
+ File classFile = ClassUtils.classNameToFile(\_rootPath, className);  
**+ assertClassfile(className, classFile);**+ FileInputStream iStream = null;  
+ int fileLength = (int) classFile.length();  
+ byte[] fileContent = new byte[fileLength];  
+ try {  
+ iStream = new FileInputStream(classFile);  
+ iStream.read(fileContent);  
+ return super.defineClass(className, fileContent, 0, fileLength);  
+ } catch (Exception e) {  
+ throw new ClassNotFoundException(e.toString());  
+ } finally {  
+ if (iStream != null) {  
+ try {  
+ iStream.close();  
+ } catch (Exception e) {  
+ }  
+ }  
+ }  
+ }  
========================================

https://issues.apache.org/jira/browse/EXTSCRIPT-98  
project reorg due to pending issues with the svn change history plugin  
  
Project: myfaces-scripting.txt  
Commit: e2504fe55933883525e442f49c54c3deae365b14  
ClassName: org.apache.myfaces.scripting.api.BaseWeaver.java  
  
+ public boolean isReloadCandidate(ReloadingMetadata reloadMeta) {  
**+ return reloadMeta != null && assertScriptingEngine(reloadMeta) && reloadMeta.isTaintedOnce();**+ }  
========================================

https://issues.apache.org/jira/browse/EXTSCRIPT-98  
project reorg due to pending issues with the svn change history plugin  
  
Project: myfaces-scripting.txt  
Commit: 1aaefa3917af7a70f7127bce8786af7926ec85b1  
ClassName: org.apache.myfaces.scripting.api.BaseWeaver.java  
  
+ public boolean isReloadCandidate(ReloadingMetadata reloadMeta) {  
**+ return reloadMeta != null && assertScriptingEngine(reloadMeta) && reloadMeta.isTaintedOnce();**+ }  
========================================

https://issues.apache.org/jira/browse/EXTSCRIPT-98  
project reorg due to pending issues with the svn change history plugin  
  
Project: myfaces-scripting.txt  
Commit: e2504fe55933883525e442f49c54c3deae365b14  
ClassName: org.apache.myfaces.scripting.core.classIdentifier.DynamicClassloader.java  
  
+ public Class loadClass(String className, boolean resolve) throws ClassNotFoundException {  
+ if (className.contains("java.lang")) {  
+ return super.loadClass(className, resolve);  
+ }  
+ File classFile = ClassUtils.classNameToFile(\_rootPath, className);  
**+ assertClassfile(className, classFile);**+ FileInputStream iStream = null;  
+ int fileLength = (int) classFile.length();  
+ byte[] fileContent = new byte[fileLength];  
+ try {  
+ iStream = new FileInputStream(classFile);  
+ iStream.read(fileContent);  
+ return super.defineClass(className, fileContent, 0, fileLength);  
+ } catch (Exception e) {  
+ throw new ClassNotFoundException(e.toString());  
+ } finally {  
+ if (iStream != null) {  
+ try {  
+ iStream.close();  
+ } catch (Exception e) {  
+ }  
+ }  
+ }  
+ }  
========================================

https://issues.apache.org/jira/browse/EXTSCRIPT-98  
project reorg due to pending issues with the svn change history plugin  
  
Project: myfaces-scripting.txt  
Commit: e2504fe55933883525e442f49c54c3deae365b14  
ClassName: org.apache.myfaces.scripting.api.BaseWeaver.java  
  
+ public Class reloadScriptingClass(Class aclass) {  
+ ReloadingMetadata metadata = getClassMap().get(aclass.getName());  
+ if (metadata == null)  
+ return aclass;  
**+ if (!assertScriptingEngine(metadata)) {**+ return null;  
+ }  
+ if (!metadata.isTainted()) {  
+ return metadata.getAClass();  
+ }  
+ synchronized (RefreshContext.COMPILE\_SYNC\_MONITOR) {  
+ if (!metadata.isTainted()) {  
+ return metadata.getAClass();  
+ }  
+ return loadScriptingClassFromFile(metadata.getSourcePath(), metadata.getFileName());  
+ }  
+ }  
========================================

Jacob packages weren't properly renamed... now fixed.  
  
Project: ode.txt  
Commit: 49a63bdddfec77a900b705c164c736e33f3acb11  
ClassName: org.apache.ode.jacob.vpu.FastSoupImpl.java  
  
+ private void removeCommGroup(CommGroupFrame groupFrame) {  
+ for (Iterator i = groupFrame.commFrames.iterator(); i.hasNext();) {  
+ CommFrame frame = (CommFrame) i.next();  
+ if (frame instanceof ObjectFrame) {  
**+ assert frame.channelFrame.objFrames.contains(frame);**+ frame.channelFrame.objFrames.remove(frame);  
+ } else {  
**+ assert frame instanceof MessageFrame;  
+ assert frame.channelFrame.msgFrames.contains(frame);**+ frame.channelFrame.msgFrames.remove(frame);  
+ }  
+ }  
+ }  
========================================

Jacob packages weren't properly renamed... now fixed.  
  
Project: ode.txt  
Commit: 49a63bdddfec77a900b705c164c736e33f3acb11  
ClassName: org.apache.ode.jacob.vpu.JacobVPU.java  
  
+ public void run() {  
**+ assert \_methodBody != null;  
+ assert \_method != null;  
+ assert \_method.getDeclaringClass()**+ .isAssignableFrom(\_methodBody.getClass());  
**+ assert \_\_activeJacobThread.get() == null;**+ if (\_\_log.isDebugEnabled()) {  
+ String dbgMsg = \_cycle + ": " + \_source;  
+ \_\_log.debug(dbgMsg);  
+ \_prefix = " ===> ";  
+ }  
+ Object[] args;  
+ SynchChannel synchChannel;  
+ if (\_method.getReturnType() != void.class) {  
+ args = new Object[\_args.length-1];  
+ System.arraycopy(\_args,0, args, 0, args.length);  
+ synchChannel = (SynchChannel)\_args[args.length];  
+ } else {  
+ args = \_args;  
+ synchChannel = null;  
+ }  
+ \_\_activeJacobThread.set(this);  
+ long ctime = System.currentTimeMillis();  
+ try {  
+ \_method.invoke(\_methodBody, args);  
+ if (synchChannel != null)  
+ synchChannel.ret();  
+ } catch (IllegalAccessException iae) {  
+ String msg = \_\_msgs.msgMethodNotAccessible(\_method.getName(), \_method.getDeclaringClass().getName());  
+ \_\_log.error(msg, iae);  
+ throw new RuntimeException(msg, iae);  
+ } catch (InvocationTargetException e) {  
+ String msg = \_\_msgs.msgClientMethodException(\_method.getName(), \_methodBody.getClass().getName());  
+ \_\_log.error(msg, e.getTargetException());  
+ throw new RuntimeException(e.getTargetException());  
+ } finally {  
+ ctime = System.currentTimeMillis() - ctime;  
+ \_statistics.totalClientTimeMs += ctime;  
+ \_\_activeJacobThread.set(null);  
+ \_prefix = null;  
+ }  
**+ assert \_\_activeJacobThread.get() == null;**+ }  
========================================

Jacob packages weren't properly renamed... now fixed.  
  
Project: ode.txt  
Commit: 49a63bdddfec77a900b705c164c736e33f3acb11  
ClassName: org.apache.ode.jacob.ML.java  
  
+ public Method getMethod(String method) {  
+ for (Iterator<Method> i = getImplementedMethods().iterator();i.hasNext();) {  
+ Method meth = i.next();  
+ if (meth.getName().equals(method)) {  
+ return meth;  
+ }  
+ }  
**+ assert !\_implementedMethods.contains(method);**+ throw new IllegalArgumentException("No such method: " + method + " in " + \_implementedMethods);  
+ }  
========================================

changeset: 102:5ce8837e22f0  
user: Developer:Michael Stahl,Svante;Reviewers:Svante,Michael,Devin  
date: Mon Nov 22 19:34:14 2010 +0100  
#bug 66# Enabling the ODF validation for ODFDOM  
  
Project: odftoolkit.txt  
Commit: cfd406095b610d615713fe66f5322dbdb62b8930  
ClassName: org.odftoolkit.odfdom.utils.ErrorHandlerStub.java  
  
+ private void logMissingConstraint(ValidationConstraint constraint, String level, int problemOccurance) {  
+ if (constraint instanceof OdfPackageConstraint) {  
+ OdfPackageConstraint pkgConstraint = (OdfPackageConstraint) constraint;  
**+ Assert.fail(problemOccurance + "time(s) missing in the document the expected ODF 1.2 Package " + level + " for " + pkgConstraint.name());**+ } else {  
+ OdfSchemaConstraint schemaConstraint = (OdfSchemaConstraint) constraint;  
**+ Assert.fail(problemOccurance + "time(s) missing in the document the expected ODF 1.2 Schema error for " + schemaConstraint.name());**+ }  
+ }  
========================================

changeset: 102:5ce8837e22f0  
user: Developer:Michael Stahl,Svante;Reviewers:Svante,Michael,Devin  
date: Mon Nov 22 19:34:14 2010 +0100  
#bug 66# Enabling the ODF validation for ODFDOM  
  
Project: odftoolkit.txt  
Commit: cfd406095b610d615713fe66f5322dbdb62b8930  
ClassName: org.odftoolkit.odfdom.utils.ErrorHandlerStub.java  
  
+ private void logUnexpectedConstraint(ValidationConstraint constraint, String level, int problemOccurance) {  
+ if (constraint instanceof OdfPackageConstraint) {  
+ OdfPackageConstraint pkgConstraint = (OdfPackageConstraint) constraint;  
**+ Assert.fail(problemOccurance + "time(s) unexpected a new ODF 1.2 Package " + level + " for " + pkgConstraint.name());**+ } else {  
+ OdfSchemaConstraint schemaConstraint = (OdfSchemaConstraint) constraint;  
**+ Assert.fail(problemOccurance + "time(s) unexpected a new ODF 1.2 Schema error for " + schemaConstraint.name());**+ }  
+ }  
========================================

changeset: 104:832ea0c43040  
user: Developer:svanteschubert,Reviewer:Devin  
date: Wed Dec 01 19:16:29 2010 +0100  
#bug 66 Enhancing package validation  
  
Project: odftoolkit.txt  
Commit: e9693aa7b805b336d9d0ff2b5afedb73a483c78c  
ClassName: org.odftoolkit.odfdom.utils.ErrorHandlerStub.java  
  
- private void logMissingConstraint(ValidationConstraint constraint, String level, int problemOccurance) {  
+ private void logMissingConstraint(ValidationConstraint constraint, String errorLevel, int problemOccurance) {  
 if (constraint instanceof OdfPackageConstraint) {  
 OdfPackageConstraint pkgConstraint = (OdfPackageConstraint) constraint;  
**- Assert.fail(problemOccurance + "time(s) missing in the document the expected ODF 1.2 Package " + level + " for " + pkgConstraint.name());  
+ Assert.fail(problemOccurance + "x time(s) was in " + getTestFilePath() + " not thrown the ODF 1.2 Package " + errorLevel + " '" + pkgConstraint.name() + "'!");** } else {  
========================================

changeset: 104:832ea0c43040  
user: Developer:svanteschubert,Reviewer:Devin  
date: Wed Dec 01 19:16:29 2010 +0100  
#bug 66 Enhancing package validation  
  
Project: odftoolkit.txt  
Commit: e9693aa7b805b336d9d0ff2b5afedb73a483c78c  
ClassName: org.odftoolkit.odfdom.utils.ErrorHandlerStub.java  
  
- private void logUnexpectedConstraint(ValidationConstraint constraint, String level, int problemOccurance) {  
+ private void logUnexpectedConstraint(ValidationConstraint constraint, String errorLevel, int problemOccurance) {  
 if (constraint instanceof OdfPackageConstraint) {  
 OdfPackageConstraint pkgConstraint = (OdfPackageConstraint) constraint;  
**- Assert.fail(problemOccurance + "time(s) unexpected a new ODF 1.2 Package " + level + " for " + pkgConstraint.name());  
+ Assert.fail(problemOccurance + "x time(s) in " + getTestFilePath() + " a new ODF 1.2 Package " + errorLevel + " '" + pkgConstraint.name() + "' was unexpected thrown!");** } else {  
========================================

Tests to show unenhanced class bugs with OpenJPA  
  
Project: openejb.txt  
Commit: d4cb3550cd34693bf56f42c9d96a216b60701469  
ClassName: org.apache.openejb.core.cmp.jpa.UnenhancedUnits.java  
  
+ public void collection() throws Exception {  
+ beginTx();  
+ OneStandalone one = new OneStandalone(1000);  
+ ManyStandalone manyA = new ManyStandalone(1);  
+ one.getMany().add(manyA);  
+ manyA.setOne(one);  
+ ManyStandalone manyB = new ManyStandalone(2);  
+ one.getMany().add(manyB);  
+ manyB.setOne(one);  
+ ManyStandalone manyC = new ManyStandalone(3);  
+ one.getMany().add(manyC);  
+ manyC.setOne(one);  
**+ assertFalse(entityManager.contains(one));  
+ assertFalse(entityManager.contains(manyA));  
+ assertFalse(entityManager.contains(manyB));  
+ assertFalse(entityManager.contains(manyC));**+ entityManager.persist(one);  
+ entityManager.persist(manyA);  
+ entityManager.persist(manyB);  
+ entityManager.persist(manyC);  
+ entityManager.flush();  
**+ assertTrue(entityManager.contains(one));  
+ assertTrue(entityManager.contains(manyA));  
+ assertTrue(entityManager.contains(manyB));  
+ assertTrue(entityManager.contains(manyC));**+ commitTx();  
+ one = null;  
+ manyA = null;  
+ manyB = null;  
+ manyC = null;  
+ beginTx();  
+ one = entityManager.find(OneStandalone.class, 1000);  
**+ assertNotNull("one is null", one);  
+ assertNotNull("one.getMany() is null", one.getMany());**+ Collection<ManyStandalone> many = one.getMany();  
**+ assertEquals(3, many.size());**+ manyA = entityManager.find(ManyStandalone.class, 1);  
**+ assertNotNull("manyA is null", manyA);**+ manyB = entityManager.find(ManyStandalone.class, 2);  
**+ assertNotNull("manyB is null", manyA);**+ manyC = entityManager.find(ManyStandalone.class, 3);  
**+ assertNotNull("manyc is null", manyA);  
+ assertNotNull("manyA.getOne() is null", manyA.getOne());  
+ assertEquals(one, manyA.getOne());  
+ assertNotNull("manyB.getOne() is null", manyB.getOne());  
+ assertEquals(one, manyB.getOne());  
+ assertNotNull("manyC.getOne() is null", manyC.getOne());  
+ assertEquals(one, manyC.getOne());  
+ assertTrue(many.contains(manyA));  
+ assertTrue(many.contains(manyB));  
+ assertTrue(many.contains(manyC));**+ commitTx();  
+ }  
========================================

Tests to show unenhanced class bugs with OpenJPA  
  
Project: openejb.txt  
Commit: d4cb3550cd34693bf56f42c9d96a216b60701469  
ClassName: org.apache.openejb.core.cmp.jpa.UnenhancedUnits.java  
  
+ public void generatedIdSubclass() throws Exception {  
+ beginTx();  
+ GeneratedSuperclass generated = new GeneratedSubclass();  
**+ assertNull("generated.getId() is not null", generated.getId());  
+ assertFalse(entityManager.contains(generated));**+ entityManager.persist(generated);  
+ entityManager.flush();  
**+ assertNotNull("generated.getId() is null", generated.getId());  
+ assertTrue(entityManager.contains(generated));**+ commitTx();  
+ }  
========================================

Make timer handles work, add some error detection and handling  
  
Project: openejb.txt  
Commit: 52ee83a6f28bc85f5aafcfa1a8b92e978442fefd  
ClassName: org.openejb.GenericEJBContainer.java  
  
+ public Timer getTimerById(Long id) {  
**+ assert timerService != null;**+ return timerService.getTimerById(id);  
+ }  
 private void setupJndi() {  
========================================

Make timer handles work, add some error detection and handling  
  
Project: openejb.txt  
Commit: 52ee83a6f28bc85f5aafcfa1a8b92e978442fefd  
ClassName: org.openejb.mdb.MDBContainer.java  
  
+ public Timer getTimerById(Long id) {  
**+ assert timerService != null;**+ return timerService.getTimerById(id);  
+ }  
 public void doStart() throws Exception {  
========================================

Tests to show unenhanced class bugs with OpenJPA  
  
Project: openejb.txt  
Commit: d4cb3550cd34693bf56f42c9d96a216b60701469  
ClassName: org.apache.openejb.core.cmp.jpa.UnenhancedUnits.java  
  
+ public void generatedId() throws Exception {  
+ beginTx();  
+ GeneratedStandalone generated = new GeneratedStandalone();  
**+ assertNull("generated.getId() is not null", generated.getId());  
+ assertFalse(entityManager.contains(generated));**+ entityManager.persist(generated);  
+ entityManager.flush();  
**+ assertNotNull("generated.getId() is null", generated.getId());  
+ assertTrue(entityManager.contains(generated));**+ commitTx();  
+ }  
========================================

Added support for in-tx cacheing back in  
This unturned a load of places that were not handling transaction contexts properly and bugs in the context itself  
Changed instance contexts so they track entrancy and can be killed as required by the spec  
Made all instance context fields final  
  
Project: openejb.txt  
Commit: 66d9cb611c3cb03dfc4fe8d29e4045bb6239c1c8  
ClassName: org.openejb.AbstractInstanceContext.java  
  
+ public void exit() {  
**+ assert isInCall();**+ callDepth--;  
+ }  
========================================

Added support for in-tx cacheing back in  
This unturned a load of places that were not handling transaction contexts properly and bugs in the context itself  
Changed instance contexts so they track entrancy and can be killed as required by the spec  
Made all instance context fields final  
  
Project: openejb.txt  
Commit: 66d9cb611c3cb03dfc4fe8d29e4045bb6239c1c8  
ClassName: org.openejb.mdb.MDBInstanceInterceptor.java  
  
@@ -80,2 +81,4 @@ public InvocationResult invoke(final Invocation invocation) throws Throwable {  
 **assert ctx.getInstance() != null: "Got a context with no instance assigned";  
+ assert !ctx.isInCall() : "Acquired a context already in an invocation";**+ ctx.setPool(pool);  
========================================

Added PrintWriter for Derby logs. Hack for setting derby.system.home. Fixed derby network startup issue  
  
Project: openejb.txt  
Commit: ec4be34e29a3230a99a2926346b403d5e3436463  
ClassName: org.apache.openejb.alt.config.RemoteServer.java  
  
+ public static void main(String[] args) {  
**+ assert args.length > 0 : "no arguments supplied: valid arguments are 'start' or 'stop'";**+ if (args[0].equalsIgnoreCase("start")){  
+ new RemoteServer().start();  
+ } else if (args[0].equalsIgnoreCase("stop")) {  
+ RemoteServer remoteServer = new RemoteServer();  
+ remoteServer.serverHasAlreadyBeenStarted = false;  
+ remoteServer.stop();  
+ } else {  
+ throw new RuntimeException("valid arguments are 'start' or 'stop'");  
+ }  
========================================

Tests to show unenhanced class bugs with OpenJPA  
  
Project: openejb.txt  
Commit: d4cb3550cd34693bf56f42c9d96a216b60701469  
ClassName: org.apache.openejb.core.cmp.jpa.UnenhancedUnits.java  
  
+ public void complexIdSubclass() throws Exception {  
+ beginTx();  
+ ComplexSuperclass complex = new ComplexSubclass();  
+ complex.firstId = "first";  
+ complex.secondId = "second";  
**+ assertFalse(entityManager.contains(complex));**+ entityManager.persist(complex);  
**+ assertTrue(entityManager.contains(complex));**+ commitTx();  
+ }  
========================================

Updated the framework to support different key types i.e. errors , warnings and failures. Now you can specify what a key represents. If you are expecting a key to throw warnings instead of failures, you can specify it like @Keys(@Key(value="foo",type=KeyType.WARNING))  
  
Project: openejb.txt  
Commit: 2192fffaf33aa254342f251a3644087e9c814e57  
ClassName: org.apache.openejb.config.rules.InvokeMethod.java  
  
@@ -72,4 +77,12 @@ public void evaluate() throws Throwable {  
 } catch (ValidationFailedException vfe) {  
- if (!expectedKeys.isEmpty()) {  
**- assertFailures(expectedKeys, vfe);**+ if (!isEmpty(expectedKeys)) {  
+ if (!expectedKeys.get(KeyType.FAILURE).isEmpty()) {  
**+ assertFailures(expectedKeys.get(KeyType.FAILURE), vfe);**+ }  
+ if (!expectedKeys.get(KeyType.WARNING).isEmpty()) {  
**+ assertWarnings(expectedKeys.get(KeyType.WARNING), vfe);**+ }  
+ if (!expectedKeys.get(KeyType.ERROR).isEmpty()) {  
**+ assertErrors(expectedKeys.get(KeyType.ERROR), vfe);**+ }  
 } else {  
========================================

Changes for JIRA OPENJPA-116 issue.  
Added a test to ensure that the EntityManager was still open before executing the getDelegate() code.  
  
Project: openjpa.txt  
Commit: 17c6580e4ce21e706513f77586372d9bf257a244  
ClassName: org.apache.openjpa.persistence.EntityManagerImpl.java  
  
 public Object getDelegate() {  
**+ \_broker.assertOpen();** return this;  
========================================

[OWB-258] InjectionPoint meta-data injection into dependent beans not work and some minor issues  
  
Project: openwebbeans.txt  
Commit: a90c7aef3fb8f02c10d7c856045616e3cceec9c3  
ClassName: org.apache.webbeans.decorator.DecoratorUtil.java  
  
@@ -60,2 +61,13 @@ public static void checkDecoratorConditions(Class<?> decoratorClazz)  
 {   
**+ Asserts.assertNotNull(decoratorClazz, "Decorator class is null");**+ Method[] methods = decoratorClazz.getDeclaredMethods();  
+ for(Method method : methods)  
+ {  
+ if(AnnotationUtil.hasMethodAnnotation(method, Produces.class))  
+ {  
+ throw new WebBeansConfigurationException("Decorator class : " + decoratorClazz + " can not have producer methods but it has one with name : " + method.getName());  
+ }  
+ }  
 Set<Type> decoratorSet = new HashSet<Type>();  
========================================

OWB-593 fix lots of AnnotatedType issues with Interceptors  
  
Project: openwebbeans.txt  
Commit: 2f880f96c6323c2e42fbf9251da0948ac88fba50  
ClassName: org.apache.webbeans.intercept.InterceptorUtil.java  
  
@@ -335,20 +335,26 @@ public boolean isLifecycleMethodInterceptor(Class<?> clazz)  
- public void checkInterceptorConditions(Class<?> clazz)  
+ public void checkInterceptorConditions(AnnotatedType annotatedType)  
 {  
**- Asserts.nullCheckForClass(clazz);  
+ Asserts.assertNotNull(annotatedType);**- Method[] methods = webBeansContext.getSecurityService().doPrivilegedGetDeclaredMethods(clazz);  
- for(Method method : methods)  
+ Set<AnnotatedMethod> methods = annotatedType.getMethods();  
+ for(AnnotatedMethod method : methods)  
 {  
- if(AnnotationUtil.hasMethodAnnotation(method, Produces.class))  
+ List<AnnotatedParameter> parms = method.getParameters();  
+ for (AnnotatedParameter parameter : parms)  
 {  
- throw new WebBeansConfigurationException("Interceptor class : " + clazz + " can not have producer methods but it has one with name : "  
- + method.getName());  
+ if (parameter.isAnnotationPresent(Produces.class))  
+ {  
+ throw new WebBeansConfigurationException("Interceptor class : " + annotatedType.getJavaClass()  
+ + " can not have producer methods but it has one with name : "  
+ + method.getJavaMember().getName());  
 }  
========================================

OWB-773 fix typo  
txs to mbenson!  
  
Project: openwebbeans.txt  
Commit: 045db9246ea2cbd9d8ea534edfdb164e9e3cd8ba  
ClassName: org.apache.webbeans.container.BeanManagerImpl.java  
  
@@ -272,3 +272,3 @@ public Context getContext(Class<? extends Annotation> scopeType)  
 {  
**- Asserts.assertNotNull(scopeType, "scopeType paramter can not be null");  
+ Asserts.assertNotNull(scopeType, "scopeType parameter can not be null");**========================================

Add a unified and optionally more constrained API for expressing filters on columns  
This is a re-opened version of:  
https://github.com/Parquet/parquet-mr/pull/412  
The idea behind this pull request is to add a way to express filters on columns using DSL that allows parquet visibility into what is being filtered and how. This visibility will allow us to make optimizations at read time, the biggest one being filtering entire row groups or pages of records without even reading them based on the statistics / metadata that is stored along with each row group or page.  
Included in this api are interfaces for user defined predicates, which must operate at the value level by may opt in to operating at the row group / page level as well. This should make this new API a superset of the `parquet.filter` package. This new api will need to be reconciled with the column filters currently in the `parquet.filter` package, but I wanted to get feedback on this first.  
A limitation in both this api and the old one is that you can't do cross-column filters, eg: columX > columnY.  
Author: Alex Levenson <alexlevenson@twitter.com>  
Closes #4 from isnotinvain/alexlevenson/filter-api and squashes the following commits:  
c1ab7e3 [Alex Levenson] Address feedback  
c1bd610 [Alex Levenson] cleanup dotString in ColumnPath  
418bfc1 [Alex Levenson] Update version, add temporary hacks for semantic enforcer  
6643bd3 [Alex Levenson] Fix some more non backward incompatible changes  
39f977f [Alex Levenson] Put a bunch of backwards compatible stuff back in, add @Deprecated  
13a02c6 [Alex Levenson] Fix compile errors, add back in overloaded getRecordReader  
f82edb7 [Alex Levenson] Merge branch 'master' into alexlevenson/filter-api  
9bd014f [Alex Levenson] clean up TODOs and reference jiras  
4cc7e87 [Alex Levenson] Add some comments  
30e3d61 [Alex Levenson] Create a common interface for both kinds of filters  
ac153a6 [Alex Levenson] Create a Statistics class for use in UDPs  
fbbf601 [Alex Levenson] refactor IncrementallyUpdatedFilterPredicateGenerator to only generate the parts that require generation  
5df47cd [Alex Levenson] Static imports of checkNotNull  
c1d1823 [Alex Levenson] address some of the minor feedback items  
67a3ba0 [Alex Levenson] update binary's toString  
3d7372b [Alex Levenson] minor fixes  
fed9531 [Alex Levenson] Add skipCurrentRecord method to clear events in thrift converter  
2e632d5 [Alex Levenson] Make Binary Serializable  
09c024f [Alex Levenson] update comments  
3169849 [Alex Levenson] fix compilation error  
0185030 [Alex Levenson] Add integration test for value level filters  
4fde18c [Alex Levenson] move to right package  
ae36b37 [Alex Levenson] Handle merge issues  
af69486 [Alex Levenson] Merge branch 'master' into alexlevenson/filter-api  
0665271 [Alex Levenson] Add tests for value inspector  
c5e3b07 [Alex Levenson] Add tests for resetter and evaluator  
29f677a [Alex Levenson] Fix scala DSL  
8897a28 [Alex Levenson] Fix some tests  
b448bee [Alex Levenson] Fix mistake in MessageColumnIO  
c8133f8 [Alex Levenson] Fix some tests  
4cf686d [Alex Levenson] more null checks  
69e683b [Alex Levenson] check all the nulls  
220a682 [Alex Levenson] more cleanup  
aad5af3 [Alex Levenson] rm generated src file from git  
5075243 [Alex Levenson] more minor cleanup  
9966713 [Alex Levenson] Hook generation into maven build  
8282725 [Alex Levenson] minor cleanup  
fea3ea9 [Alex Levenson] minor cleanup  
9e35406 [Alex Levenson] move statistics filter  
c52750c [Alex Levenson] finish moving things around  
97a6bfd [Alex Levenson] Move things around pt2  
843b9fe [Alex Levenson] Move some files around pt 1  
5eedcc0 [Alex Levenson] turn off dictionary support for AtomicConverter  
541319e [Alex Levenson] various cleanup and fixes  
08e9638 [Alex Levenson] rm ColumnPathUtil  
bfe6795 [Alex Levenson] Add type bounds to FilterApi  
6c831ab [Alex Levenson] don't double log exception in SerializationUtil  
a7a58d1 [Alex Levenson] use ColumnPath instead of String  
8f11a6b [Alex Levenson] Move ColumnPath and Canonicalizer to parquet-common  
9164359 [Alex Levenson] stash  
abc2be2 [Alex Levenson] Add null handling to record filters -- this impl is still broken though  
90ba8f7 [Alex Levenson] Update Serialization Util  
0a261f1 [Alex Levenson] Add compression in SerializationUtil  
f1278be [Alex Levenson] Add comment, fix tests  
cbd1a85 [Alex Levenson] Replace some specialization with generic views  
e496cbf [Alex Levenson] Fix short circuiting in StatisticsFilter  
db6b32d [Alex Levenson] Address some comments, fix constructor in ParquetReader  
fd6f44d [Alex Levenson] Fix semver backward compat  
2fdd304 [Alex Levenson] Some more cleanup  
d34fb89 [Alex Levenson] Cleanup some TODOs  
544499c [Alex Levenson] stash  
7b32016 [Alex Levenson] Merge branch 'master' into alexlevenson/filter-api  
0e31251 [Alex Levenson] First pass at values filter, needs reworking  
470e409 [Alex Levenson] fix java6/7 bug, minor cleanup  
ee7b221 [Alex Levenson] more InputFormat tests  
5ef849e [Alex Levenson] Add guards for not specifying both kinds of filter  
0186b1f [Alex Levenson] Add logging to ParquetInputFormat and tests for configuration  
a622648 [Alex Levenson] cleanup imports  
9b1ea88 [Alex Levenson] Add tests for statistics filter  
d517373 [Alex Levenson] tests for filter validator  
b25fc44 [Alex Levenson] small cleanup of filter validator  
32067a1 [Alex Levenson] add test for collapse logical nots  
1efc198 [Alex Levenson] Add tests for invert filter predicate  
046b106 [Alex Levenson] some more fixes  
d3c4d7a [Alex Levenson] fix some more types, add in test for SerializationUtil  
cc51274 [Alex Levenson] fix generics in FilterPredicateInverter  
ea08349 [Alex Levenson] First pass at rowgroup filter, needs testing  
156d91b [Alex Levenson] Add runtime type checker  
4dfb4f2 [Alex Levenson] Add serialization util  
8f80b20 [Alex Levenson] update comment  
7c25121 [Alex Levenson] Add class to Column struct  
58f1190 [Alex Levenson] Remove filterByUniqueValues  
7f20de6 [Alex Levenson] rename user predicates  
af14b42 [Alex Levenson] Update dsl  
04409c5 [Alex Levenson] Add generic types into Visitor  
ba42884 [Alex Levenson] rm getClassName  
  
65f8af9 [Alex Levenson] Add in support for user defined predicates on columns  
6926337 [Alex Levenson] Add explicit tokens for notEq, ltEq, gtEq  
667ec9f [Alex Levenson] remove test for collapsing double negation  
db2f71a [Alex Levenson] rename FilterPredicatesTest  
a0a0533 [Alex Levenson] Address first round of comments  
b2bca94 [Alex Levenson] Add scala DSL and tests  
bedda87 [Alex Levenson] Add tests for FilterPredicate building  
238cbbe [Alex Levenson] Add scala dsl  
39f7b24 [Alex Levenson] add scala mvn boilerplate  
2ec71a7 [Alex Levenson] Add predicate API  
  
Project: parquet-mr.txt  
Commit: ad32bf0fd111ab473ad1080cde11de39e3c5a67f  
ClassName: parquet.filter2.predicate.ValidTypeMap.java  
  
**+ public static <T extends Comparable<T>> void assertTypeValid(Column<T> foundColumn, PrimitiveTypeName primitiveType, OriginalType originalType) {**+ Class<T> foundColumnType = foundColumn.getColumnType();  
+ ColumnPath columnPath = foundColumn.getColumnPath();  
+ Set<FullTypeDescriptor> validTypeDescriptors = classToParquetType.get(foundColumnType);  
+ FullTypeDescriptor typeInFileMetaData = new FullTypeDescriptor(primitiveType, originalType);  
+ if (validTypeDescriptors == null) {  
+ StringBuilder message = new StringBuilder();  
+ message  
+ .append("Column ")  
+ .append(columnPath.toDotString())  
+ .append(" was declared as type: ")  
+ .append(foundColumnType.getName())  
+ .append(" which is not supported in FilterPredicates.");  
+ Set<Class<?>> supportedTypes = parquetTypeToClass.get(typeInFileMetaData);  
+ if (supportedTypes != null) {  
+ message  
+ .append(" Supported types for this column are: ")  
+ .append(supportedTypes);  
+ } else {  
+ message.append(" There are no supported types for columns of " + typeInFileMetaData);  
+ }  
+ throw new IllegalArgumentException(message.toString());  
+ }  
+ if (!validTypeDescriptors.contains(typeInFileMetaData)) {  
+ StringBuilder message = new StringBuilder();  
+ message  
+ .append("FilterPredicate column: ")  
+ .append(columnPath.toDotString())  
+ .append("'s declared type (")  
+ .append(foundColumnType.getName())  
+ .append(") does not match the schema found in file metadata. Column ")  
+ .append(columnPath.toDotString())  
+ .append(" is of type: ")  
+ .append(typeInFileMetaData)  
+ .append("\nValid types for this column are: ")  
+ .append(parquetTypeToClass.get(typeInFileMetaData));  
+ throw new IllegalArgumentException(message.toString());  
+ }  
+ }  
========================================

PARQUET-84: Avoid reading rowgroup metadata in memory on the client side.  
This will improve reading big datasets with a large schema (thousands of columns)  
Instead rowgroup metadata can be read in the tasks where each tasks reads only the metadata of the file it's reading  
Author: julien <julien@twitter.com>  
Closes #45 from julienledem/skip\_reading\_row\_groups and squashes the following commits:  
ccdd08c [julien] fix parquet-hive  
24a2050 [julien] Merge branch 'master' into skip\_reading\_row\_groups  
3d7e35a [julien] adress review feedback  
5b6bd1b [julien] more tests  
323d254 [julien] sdd unit tests  
f599259 [julien] review feedback  
fb11f02 [julien] fix backward compatibility check  
2c20b46 [julien] cleanup readFooters methods  
3da37d8 [julien] fix read summary  
ab95a45 [julien] cleanup  
4d16df3 [julien] implement task side metadata  
9bb8059 [julien] first stab at integrating skipping row groups  
  
Project: parquet-mr.txt  
Commit: 5dafd127f3de7c516ce9c1b7329087a01ab2fc57  
ClassName: parquet.hadoop.ParquetInputFormat.java  
  
+ static List<SplitInfo> generateSplitInfo(  
+ List<BlockMetaData> rowGroupBlocks,  
+ BlockLocation[] hdfsBlocksArray,  
+ long minSplitSize, long maxSplitSize) {  
+ List<SplitInfo> splitRowGroups;  
+ if (maxSplitSize < minSplitSize || maxSplitSize < 0 || minSplitSize < 0) {  
+ throw new ParquetDecodingException("maxSplitSize and minSplitSize should be positive and max should be greater or equal to the minSplitSize: maxSplitSize = " + maxSplitSize + "; minSplitSize is " + minSplitSize);  
+ }  
+ HDFSBlocks hdfsBlocks = new HDFSBlocks(hdfsBlocksArray);  
+ hdfsBlocks.checkBelongingToANewHDFSBlock(rowGroupBlocks.get(0));  
+ SplitInfo currentSplit = new SplitInfo(hdfsBlocks.getCurrentBlock());  
+ splitRowGroups = new ArrayList<SplitInfo>();  
**+ checkSorted(rowGroupBlocks);//assert row groups are sorted**+ for (BlockMetaData rowGroupMetadata : rowGroupBlocks) {  
+ if ((hdfsBlocks.checkBelongingToANewHDFSBlock(rowGroupMetadata)  
+ && currentSplit.getCompressedByteSize() >= minSplitSize  
+ && currentSplit.getCompressedByteSize() > 0)  
+ || currentSplit.getCompressedByteSize() >= maxSplitSize) {  
+ splitRowGroups.add(currentSplit);//finish previous split  
+ currentSplit = new SplitInfo(hdfsBlocks.getCurrentBlock());  
+ }  
+ currentSplit.addRowGroup(rowGroupMetadata);  
+ }  
+ if (currentSplit.getRowGroupCount() > 0) {  
+ splitRowGroups.add(currentSplit);  
+ }  
+ return splitRowGroups;  
+ }  
========================================

Add a unified and optionally more constrained API for expressing filters on columns  
This is a re-opened version of:  
https://github.com/Parquet/parquet-mr/pull/412  
The idea behind this pull request is to add a way to express filters on columns using DSL that allows parquet visibility into what is being filtered and how. This visibility will allow us to make optimizations at read time, the biggest one being filtering entire row groups or pages of records without even reading them based on the statistics / metadata that is stored along with each row group or page.  
Included in this api are interfaces for user defined predicates, which must operate at the value level by may opt in to operating at the row group / page level as well. This should make this new API a superset of the `parquet.filter` package. This new api will need to be reconciled with the column filters currently in the `parquet.filter` package, but I wanted to get feedback on this first.  
A limitation in both this api and the old one is that you can't do cross-column filters, eg: columX > columnY.  
Author: Alex Levenson <alexlevenson@twitter.com>  
Closes #4 from isnotinvain/alexlevenson/filter-api and squashes the following commits:  
c1ab7e3 [Alex Levenson] Address feedback  
c1bd610 [Alex Levenson] cleanup dotString in ColumnPath  
418bfc1 [Alex Levenson] Update version, add temporary hacks for semantic enforcer  
6643bd3 [Alex Levenson] Fix some more non backward incompatible changes  
39f977f [Alex Levenson] Put a bunch of backwards compatible stuff back in, add @Deprecated  
13a02c6 [Alex Levenson] Fix compile errors, add back in overloaded getRecordReader  
f82edb7 [Alex Levenson] Merge branch 'master' into alexlevenson/filter-api  
9bd014f [Alex Levenson] clean up TODOs and reference jiras  
4cc7e87 [Alex Levenson] Add some comments  
30e3d61 [Alex Levenson] Create a common interface for both kinds of filters  
ac153a6 [Alex Levenson] Create a Statistics class for use in UDPs  
fbbf601 [Alex Levenson] refactor IncrementallyUpdatedFilterPredicateGenerator to only generate the parts that require generation  
5df47cd [Alex Levenson] Static imports of checkNotNull  
c1d1823 [Alex Levenson] address some of the minor feedback items  
67a3ba0 [Alex Levenson] update binary's toString  
3d7372b [Alex Levenson] minor fixes  
fed9531 [Alex Levenson] Add skipCurrentRecord method to clear events in thrift converter  
2e632d5 [Alex Levenson] Make Binary Serializable  
09c024f [Alex Levenson] update comments  
3169849 [Alex Levenson] fix compilation error  
0185030 [Alex Levenson] Add integration test for value level filters  
4fde18c [Alex Levenson] move to right package  
ae36b37 [Alex Levenson] Handle merge issues  
af69486 [Alex Levenson] Merge branch 'master' into alexlevenson/filter-api  
0665271 [Alex Levenson] Add tests for value inspector  
c5e3b07 [Alex Levenson] Add tests for resetter and evaluator  
29f677a [Alex Levenson] Fix scala DSL  
8897a28 [Alex Levenson] Fix some tests  
b448bee [Alex Levenson] Fix mistake in MessageColumnIO  
c8133f8 [Alex Levenson] Fix some tests  
4cf686d [Alex Levenson] more null checks  
69e683b [Alex Levenson] check all the nulls  
220a682 [Alex Levenson] more cleanup  
aad5af3 [Alex Levenson] rm generated src file from git  
5075243 [Alex Levenson] more minor cleanup  
9966713 [Alex Levenson] Hook generation into maven build  
8282725 [Alex Levenson] minor cleanup  
fea3ea9 [Alex Levenson] minor cleanup  
9e35406 [Alex Levenson] move statistics filter  
c52750c [Alex Levenson] finish moving things around  
97a6bfd [Alex Levenson] Move things around pt2  
843b9fe [Alex Levenson] Move some files around pt 1  
5eedcc0 [Alex Levenson] turn off dictionary support for AtomicConverter  
541319e [Alex Levenson] various cleanup and fixes  
08e9638 [Alex Levenson] rm ColumnPathUtil  
bfe6795 [Alex Levenson] Add type bounds to FilterApi  
6c831ab [Alex Levenson] don't double log exception in SerializationUtil  
a7a58d1 [Alex Levenson] use ColumnPath instead of String  
8f11a6b [Alex Levenson] Move ColumnPath and Canonicalizer to parquet-common  
9164359 [Alex Levenson] stash  
abc2be2 [Alex Levenson] Add null handling to record filters -- this impl is still broken though  
90ba8f7 [Alex Levenson] Update Serialization Util  
0a261f1 [Alex Levenson] Add compression in SerializationUtil  
f1278be [Alex Levenson] Add comment, fix tests  
cbd1a85 [Alex Levenson] Replace some specialization with generic views  
e496cbf [Alex Levenson] Fix short circuiting in StatisticsFilter  
db6b32d [Alex Levenson] Address some comments, fix constructor in ParquetReader  
fd6f44d [Alex Levenson] Fix semver backward compat  
2fdd304 [Alex Levenson] Some more cleanup  
d34fb89 [Alex Levenson] Cleanup some TODOs  
544499c [Alex Levenson] stash  
7b32016 [Alex Levenson] Merge branch 'master' into alexlevenson/filter-api  
0e31251 [Alex Levenson] First pass at values filter, needs reworking  
470e409 [Alex Levenson] fix java6/7 bug, minor cleanup  
ee7b221 [Alex Levenson] more InputFormat tests  
5ef849e [Alex Levenson] Add guards for not specifying both kinds of filter  
0186b1f [Alex Levenson] Add logging to ParquetInputFormat and tests for configuration  
a622648 [Alex Levenson] cleanup imports  
9b1ea88 [Alex Levenson] Add tests for statistics filter  
d517373 [Alex Levenson] tests for filter validator  
b25fc44 [Alex Levenson] small cleanup of filter validator  
32067a1 [Alex Levenson] add test for collapse logical nots  
1efc198 [Alex Levenson] Add tests for invert filter predicate  
046b106 [Alex Levenson] some more fixes  
d3c4d7a [Alex Levenson] fix some more types, add in test for SerializationUtil  
cc51274 [Alex Levenson] fix generics in FilterPredicateInverter  
ea08349 [Alex Levenson] First pass at rowgroup filter, needs testing  
156d91b [Alex Levenson] Add runtime type checker  
4dfb4f2 [Alex Levenson] Add serialization util  
8f80b20 [Alex Levenson] update comment  
7c25121 [Alex Levenson] Add class to Column struct  
58f1190 [Alex Levenson] Remove filterByUniqueValues  
7f20de6 [Alex Levenson] rename user predicates  
af14b42 [Alex Levenson] Update dsl  
04409c5 [Alex Levenson] Add generic types into Visitor  
ba42884 [Alex Levenson] rm getClassName  
  
65f8af9 [Alex Levenson] Add in support for user defined predicates on columns  
6926337 [Alex Levenson] Add explicit tokens for notEq, ltEq, gtEq  
667ec9f [Alex Levenson] remove test for collapsing double negation  
db2f71a [Alex Levenson] rename FilterPredicatesTest  
a0a0533 [Alex Levenson] Address first round of comments  
b2bca94 [Alex Levenson] Add scala DSL and tests  
bedda87 [Alex Levenson] Add tests for FilterPredicate building  
238cbbe [Alex Levenson] Add scala dsl  
39f7b24 [Alex Levenson] add scala mvn boilerplate  
2ec71a7 [Alex Levenson] Add predicate API  
  
Project: parquet-mr.txt  
Commit: ad32bf0fd111ab473ad1080cde11de39e3c5a67f  
ClassName: parquet.filter2.predicate.SchemaCompatibilityValidator.java  
  
+ private <T extends Comparable<T>> void validateColumn(Column<T> column) {  
+ ColumnPath path = column.getColumnPath();  
+ Class<?> alreadySeen = columnTypesEncountered.get(path);  
+ if (alreadySeen != null && !alreadySeen.equals(column.getColumnType())) {  
+ throw new IllegalArgumentException("Column: "  
+ + path.toDotString()  
+ + " was provided with different types in the same predicate."  
+ + " Found both: (" + alreadySeen + ", " + column.getColumnType() + ")");  
+ }  
+ columnTypesEncountered.put(path, column.getColumnType());  
+ ColumnDescriptor descriptor = getColumnDescriptor(path);  
+ if (descriptor.getMaxRepetitionLevel() > 0) {  
+ throw new IllegalArgumentException("FilterPredicates do not currently support repeated columns. "  
+ + "Column " + path.toDotString() + " is repeated.");  
+ }  
**+ ValidTypeMap.assertTypeValid(column, descriptor.getType(), originalTypes.get(path));**+ }  
========================================

Rename package from org.apache.hadoop.hbase.index.\* to org.apache.phoenix.index.\* to fix classloader issue causing mutable index performance regression - https://issues.apache.org/jira/browse/PHOENIX-38  
  
Project: phoenix.txt  
Commit: bbacf6e03045bab692aecbea00a4b63ee7593e73  
ClassName: org.apache.phoenix.hbase.index.builder.IndexBuildManager.java  
  
+ public Collection<Pair<Mutation, byte[]>> getIndexUpdate(  
+ MiniBatchOperationInProgress<Pair<Mutation, Integer>> miniBatchOp,  
+ Collection<? extends Mutation> mutations) throws Throwable {  
+ this.delegate.batchStarted(miniBatchOp);  
+ TaskBatch<Collection<Pair<Mutation, byte[]>>> tasks =  
+ new TaskBatch<Collection<Pair<Mutation, byte[]>>>(mutations.size());  
+ for (final Mutation m : mutations) {  
+ tasks.add(new Task<Collection<Pair<Mutation, byte[]>>>() {  
+ @Override  
+ public Collection<Pair<Mutation, byte[]>> call() throws IOException {  
+ return delegate.getIndexUpdate(m);  
+ }  
+ });  
+ }  
+ List<Collection<Pair<Mutation, byte[]>>> allResults = null;  
+ try {  
+ allResults = pool.submitUninterruptible(tasks);  
+ } catch (CancellationException e) {  
+ throw e;  
+ } catch (ExecutionException e) {  
+ LOG.error("Found a failed index update!");  
+ throw e.getCause();  
+ }  
+ Collection<Pair<Mutation, byte[]>> results = new ArrayList<Pair<Mutation, byte[]>>();  
+ for (Collection<Pair<Mutation, byte[]>> result : allResults) {  
**+ assert result != null : "Found an unsuccessful result, but didn't propagate a failure earlier";**+ results.addAll(result);  
+ }  
+ return results;  
+ }  
========================================

Rename package from org.apache.hadoop.hbase.index.\* to org.apache.phoenix.index.\* to fix classloader issue causing mutable index performance regression - https://issues.apache.org/jira/browse/PHOENIX-38  
  
Project: phoenix.txt  
Commit: bbacf6e03045bab692aecbea00a4b63ee7593e73  
ClassName: org.apache.phoenix.hbase.index.covered.data.LocalTable.java  
  
+ public Result getCurrentRowState(Mutation m, Collection<? extends ColumnReference> columns)  
+ throws IOException {  
+ byte[] row = m.getRow();  
+ Scan s = IndexManagementUtil.newLocalStateScan(Collections.singletonList(columns));  
+ s.setStartRow(row);  
+ s.setStopRow(row);  
+ HRegion region = this.env.getRegion();  
+ RegionScanner scanner = region.getScanner(s);  
+ List<KeyValue> kvs = new ArrayList<KeyValue>(1);  
+ boolean more = scanner.next(kvs);  
**+ assert !more : "Got more than one result when scanning" + " a single row in the primary table!";**+ Result r = new Result(kvs);  
+ scanner.close();  
+ return r;  
+ }  
========================================

PHOENIX-1344 fix duplicit values in NTH\_VALUE function (Vaclav Loffelmann)  
  
Project: phoenix.txt  
Commit: 7835f41efd74664f73bed866d40a958d43e0ace2  
ClassName: org.apache.phoenix.end2end.NthValueFunctionIT.java  
  
+ public void nonUniqueValuesInOrderNextValueAsc() throws Exception {  
+ Connection conn = DriverManager.getConnection(getUrl());  
+ String ddl = "CREATE TABLE IF NOT EXISTS nthValue "  
+ + "(id INTEGER NOT NULL PRIMARY KEY, page\_id UNSIGNED\_LONG,"  
+ + " dates INTEGER, val INTEGER)";  
+ conn.createStatement().execute(ddl);  
+ conn.createStatement().execute("UPSERT INTO nthValue (id, page\_id, dates, val) VALUES (2, 8, 0, 7)");  
+ conn.createStatement().execute("UPSERT INTO nthValue (id, page\_id, dates, val) VALUES (3, 8, 1, 9)");  
+ conn.createStatement().execute("UPSERT INTO nthValue (id, page\_id, dates, val) VALUES (4, 8, 2, 4)");  
+ conn.createStatement().execute("UPSERT INTO nthValue (id, page\_id, dates, val) VALUES (5, 8, 2, 2)");  
+ conn.createStatement().execute("UPSERT INTO nthValue (id, page\_id, dates, val) VALUES (6, 8, 3, 3)");  
+ conn.createStatement().execute("UPSERT INTO nthValue (id, page\_id, dates, val) VALUES (7, 8, 3, 5)");  
+ conn.commit();  
+ ResultSet rs = conn.createStatement().executeQuery("SELECT NTH\_VALUE(val, 5) WITHIN GROUP (ORDER BY dates ASC) FROM nthValue GROUP BY page\_id");  
**+ assertTrue(rs.next());  
+ assertInIntArray(new int[]{3, 5}, rs.getInt(1));  
+ assertFalse(rs.next());**+ }  
========================================

PHOENIX-127 Query with pk IS NULL OR pk IN (1,2) return incorrect results (JamesTaylor)  
  
Project: phoenix.txt  
Commit: b5c5381341cf28dc73df64c2ff61197be9e7bf13  
ClassName: org.apache.phoenix.end2end.QueryIT.java  
  
 **private void assertValuesEqualsResultSet(ResultSet rs, List<List<Object>> expectedResults) throws SQLException {**- Set<List<Object>> expectedResultsSet = Sets.newHashSet(expectedResults);  
+ int expectedCount = expectedResults.size();  
 int count = 0;  
- while (rs.next()) {  
- List<Object> results = Lists.newArrayList();  
+ List<List<Object>> actualResults = Lists.newArrayList();  
+ List<Object> errorResult = null;  
+ while (rs.next() && errorResult == null) {  
+ List<Object> result = Lists.newArrayList();  
 for (int i = 0; i < rs.getMetaData().getColumnCount(); i++) {  
- results.add(rs.getObject(i+1));  
+ result.add(rs.getObject(i+1));  
 }  
**- assertTrue("Expected to find " + results + " in results: " + expectedResults, expectedResultsSet.contains(results));**+ if (!expectedResults.contains(result)) {  
+ errorResult = result;  
+ }  
+ actualResults.add(result);  
 count++;  
 }  
**- assertEquals(count, expectedResults.size());  
+ assertTrue("Could not find " + errorResult + " in expected results: " + expectedResults + " with actual results: " + actualResults, errorResult == null);  
+ assertEquals(count, expectedCount);** }  
========================================

Rename package from org.apache.hadoop.hbase.index.\* to org.apache.phoenix.index.\* to fix classloader issue causing mutable index performance regression - https://issues.apache.org/jira/browse/PHOENIX-38  
  
Project: phoenix.txt  
Commit: bbacf6e03045bab692aecbea00a4b63ee7593e73  
ClassName: org.apache.phoenix.hbase.index.Indexer.java  
  
+ public void preBatchMutateWithExceptions(ObserverContext<RegionCoprocessorEnvironment> c,  
+ MiniBatchOperationInProgress<Pair<Mutation, Integer>> miniBatchOp) throws Throwable {  
+ Map<ImmutableBytesPtr, MultiMutation> mutations =  
+ new HashMap<ImmutableBytesPtr, MultiMutation>();  
+ boolean durable = false;  
+ for (int i = 0; i < miniBatchOp.size(); i++) {  
+ WALEdit edit = miniBatchOp.getWalEdit(i);  
+ if (edit != null) {  
+ KeyValue kv = edit.getKeyValues().remove(0);  
**+ assert kv == BATCH\_MARKER : "Expected batch marker from the WALEdit, but got: " + kv;**+ }  
+ Pair<Mutation, Integer> op = miniBatchOp.getOperation(i);  
+ Mutation m = op.getFirst();  
+ if (!this.builder.isEnabled(m)) {  
+ continue;  
+ }  
+ if(!durable){  
+ durable = m.getDurability() != Durability.SKIP\_WAL;  
+ }  
+ ImmutableBytesPtr row = new ImmutableBytesPtr(m.getRow());  
+ MultiMutation stored = mutations.get(row);  
+ if (stored == null) {  
+ stored = new MultiMutation(row, m.getWriteToWAL());  
+ mutations.put(row, stored);  
+ }  
+ stored.addAll(m);  
+ }  
+ if (mutations.entrySet().size() == 0) {  
+ return;  
+ }  
+ WALEdit edit = miniBatchOp.getWalEdit(0);  
+ Collection<Pair<Mutation, byte[]>> indexUpdates =  
+ this.builder.getIndexUpdate(miniBatchOp, mutations.values());  
+ if (doPre(indexUpdates, edit, durable)) {  
+ takeUpdateLock("batch mutation");  
+ }  
+ }  
========================================

PHOENIX-1366 ORDINAL\_POSITION incorrect for multi-tenant table over tenant-specific connection (Bruno Dumon)  
  
Project: phoenix.txt  
Commit: 53c1fd7884a5eeaecf882cb199f7a25acb2d8212  
ClassName: org.apache.phoenix.end2end.TenantSpecificTablesDDLIT.java  
  
**@@ -589,2 +592,8 @@ private void assertColumnMetaData(ResultSet rs, String schema, String table, Str** }  
**+ private void assertColumnMetaData(ResultSet rs, String schema, String table, String column, int ordinalPosition)**+ throws SQLException {  
========================================

PHOENIX-1344 fix duplicit values in NTH\_VALUE function (Vaclav Loffelmann)  
  
Project: phoenix.txt  
Commit: 7835f41efd74664f73bed866d40a958d43e0ace2  
ClassName: org.apache.phoenix.end2end.NthValueFunctionIT.java  
  
+ public void nonUniqueValuesInOrderByAsc() throws Exception {  
+ Connection conn = DriverManager.getConnection(getUrl());  
+ String ddl = "CREATE TABLE IF NOT EXISTS nthValue "  
+ + "(id INTEGER NOT NULL PRIMARY KEY, page\_id UNSIGNED\_LONG,"  
+ + " dates INTEGER, val INTEGER)";  
+ conn.createStatement().execute(ddl);  
+ conn.createStatement().execute("UPSERT INTO nthValue (id, page\_id, dates, val) VALUES (2, 8, 1, 7)");  
+ conn.createStatement().execute("UPSERT INTO nthValue (id, page\_id, dates, val) VALUES (3, 8, 2, 9)");  
+ conn.createStatement().execute("UPSERT INTO nthValue (id, page\_id, dates, val) VALUES (4, 8, 2, 4)");  
+ conn.createStatement().execute("UPSERT INTO nthValue (id, page\_id, dates, val) VALUES (5, 8, 2, 2)");  
+ conn.createStatement().execute("UPSERT INTO nthValue (id, page\_id, dates, val) VALUES (6, 8, 3, 3)");  
+ conn.commit();  
+ ResultSet rs = conn.createStatement().executeQuery("SELECT NTH\_VALUE(val, 3) WITHIN GROUP (ORDER BY dates ASC) FROM nthValue GROUP BY page\_id");  
**+ assertTrue(rs.next());  
+ assertInIntArray(new int[]{2, 4, 9}, rs.getInt(1));**========================================

PHOENIX-1344 fix duplicit values in NTH\_VALUE function (Vaclav Loffelmann)  
  
Project: phoenix.txt  
Commit: 7835f41efd74664f73bed866d40a958d43e0ace2  
ClassName: org.apache.phoenix.end2end.NthValueFunctionIT.java  
  
+ public void ignoreNullValues() throws Exception {  
+ Connection conn = DriverManager.getConnection(getUrl());  
+ String ddl = "CREATE TABLE IF NOT EXISTS nth\_test\_table "  
+ + "(id INTEGER NOT NULL, page\_id UNSIGNED\_LONG,"  
+ + " dates BIGINT NOT NULL, \"value\" BIGINT NULL CONSTRAINT pk PRIMARY KEY (id, dates))";  
+ conn.createStatement().execute(ddl);  
+ conn.createStatement().execute("UPSERT INTO nth\_test\_table (id, page\_id, dates, \"value\") VALUES (1, 8, 1, 1)");  
+ conn.createStatement().execute("UPSERT INTO nth\_test\_table (id, page\_id, dates, \"value\") VALUES (2, 8, 2, NULL)");  
+ conn.createStatement().execute("UPSERT INTO nth\_test\_table (id, page\_id, dates, \"value\") VALUES (3, 8, 3, NULL)");  
+ conn.createStatement().execute("UPSERT INTO nth\_test\_table (id, page\_id, dates, \"value\") VALUES (5, 8, 4, 4)");  
+ conn.createStatement().execute("UPSERT INTO nth\_test\_table (id, page\_id, dates, \"value\") VALUES (4, 8, 5, 5)");  
+ conn.commit();  
+ ResultSet rs = conn.createStatement().executeQuery(  
+ "SELECT NTH\_VALUE(\"value\", 2) WITHIN GROUP (ORDER BY dates DESC) FROM nth\_test\_table GROUP BY page\_id");  
**+ assertTrue(rs.next());  
+ assertEquals(rs.getLong(1), 4);  
+ assertFalse(rs.next());**+ }  
========================================

PHOENIX-1344 fix duplicit values in NTH\_VALUE function (Vaclav Loffelmann)  
  
Project: phoenix.txt  
Commit: 7835f41efd74664f73bed866d40a958d43e0ace2  
ClassName: org.apache.phoenix.end2end.NthValueFunctionIT.java  
  
**+ private void assertInIntArray(int[] should, int actualValue) {**+ ArrayList<Integer> shouldList = new ArrayList<Integer>();  
+ for (int i: should) {  
+ shouldList.add(i);  
+ }  
**+ assertTrue(shouldList.contains(actualValue));**+ }  
========================================

PHOENIX-1344 fix duplicit values in NTH\_VALUE function (Vaclav Loffelmann)  
  
Project: phoenix.txt  
Commit: 7835f41efd74664f73bed866d40a958d43e0ace2  
ClassName: org.apache.phoenix.end2end.NthValueFunctionIT.java  
  
+ public void nonUniqueValuesInOrderByAscSkipDuplicit() throws Exception {  
+ Connection conn = DriverManager.getConnection(getUrl());  
+ String ddl = "CREATE TABLE IF NOT EXISTS nthValue "  
+ + "(id INTEGER NOT NULL PRIMARY KEY, page\_id UNSIGNED\_LONG,"  
+ + " dates INTEGER, val INTEGER)";  
+ conn.createStatement().execute(ddl);  
+ conn.createStatement().execute("UPSERT INTO nthValue (id, page\_id, dates, val) VALUES (2, 8, 1, 7)");  
+ conn.createStatement().execute("UPSERT INTO nthValue (id, page\_id, dates, val) VALUES (3, 8, 2, 9)");  
+ conn.createStatement().execute("UPSERT INTO nthValue (id, page\_id, dates, val) VALUES (4, 8, 2, 4)");  
+ conn.createStatement().execute("UPSERT INTO nthValue (id, page\_id, dates, val) VALUES (5, 8, 2, 2)");  
+ conn.createStatement().execute("UPSERT INTO nthValue (id, page\_id, dates, val) VALUES (6, 8, 3, 3)");  
+ conn.commit();  
+ ResultSet rs = conn.createStatement().executeQuery("SELECT NTH\_VALUE(val, 5) WITHIN GROUP (ORDER BY dates ASC) FROM nthValue GROUP BY page\_id");  
**+ assertTrue(rs.next());  
+ assertEquals(3, rs.getInt(1));  
+ assertFalse(rs.next());**+ }  
========================================

Fix issues with MenuBar; MenuButton is now broken and will be fixed in a subsequent update.  
  
Project: pivot.txt  
Commit: 4bafb4f7b7571d1290498da1400a4c163736c9d3  
ClassName: org.apache.pivot.wtk.Window.java  
  
 protected void setFocusDescendant(Component focusDescendant) {  
**+ assert(focusDescendant == null**+ || focusDescendant.getWindow() == this);  
 this.focusDescendant = focusDescendant;  
========================================

Various fixes to ListSelection.  
  
Project: pivot.txt  
Commit: 0ef49561cfd272cd474f3178f6cb72285fa6fa63  
ClassName: org.apache.pivot.wtk.ListSelection.java  
  
 public Sequence<Span> addRange(int start, int end) {  
**- assert(start > 0);  
+ assert(start >= 0);  
 assert(end >= start);**========================================

Fix some issues with focus management:  
- Component#isFocusable() now checks for showing and enabled, not showing and !blocked  
- Container#requestFocus() now only checks for showing, not showing and !blocked  
**- Window#restoreFocus() now only asserts showing, not showing and !blocked**- ApplicationContext.DisplayHost#processKeyEvent() doesn't propagate events to the focused component if it is blocked  
Partially implement multiple selection support in ListSelection class; add demo app to test selection state.  
  
Project: pivot.txt  
Commit: f18dd9fde76163cecc240d20879b43faeb873ee8  
ClassName: org.apache.pivot.wtk.Window.java  
  
 protected void restoreFocus() {  
**- assert(isShowing()**- && !isBlocked());  
**+ assert(isShowing());**========================================

Continued TextArea-related development; minor cursor-related bug fixes.  
  
Project: pivot.txt  
Commit: cec4087c7082b5cd937d3e2fabdfe0c930f2f479  
ClassName: pivot.wtk.skin.TextAreaSkin.java  
  
@@ -121,4 +122,4 @@ protected void setSize(int width, int height) {  
 **assert(height >= 0);  
- assert(maximumWidth == -1**- || width <= maximumWidth);  
**+ assert(breakWidth == -1**+ || width <= breakWidth);  
========================================

fix eclipse warnings, last part  
  
Project: pivot.txt  
Commit: be0cb562062ce3abee7614fad5d3dc733aaa055b  
ClassName: org.apache.pivot.wtk.skin.TextPaneSkinNodeView.java  
  
- public void repaint(int x, int y, int width, int height) {  
**- assert(width >= 0);  
- assert(height >= 0);**+ public void repaint(int xArgument, int yArgument, int widthArgument, int heightArgument) {  
**+ assert(widthArgument >= 0);  
+ assert(heightArgument >= 0);** if (parent != null) {  
- parent.repaint(x + this.x, y + this.y, width, height);  
+ parent.repaint(xArgument + this.x, yArgument + this.y, widthArgument, heightArgument);  
 }  
========================================

Fix bug in restoreFocus(); move logic for blocking owner window from TerraSheetSkin to Sheet.  
  
Project: pivot.txt  
Commit: e3c9fc79f118878807457a041d31a9fe02f94748  
ClassName: org.apache.pivot.wtk.Window.java  
  
 protected void restoreFocus() {  
- if (focusDescendant != null  
- && isAncestor(focusDescendant)  
- && !focusDescendant.isBlocked()  
- && focusDescendant.isShowing()) {  
**+ assert(isShowing()**+ && !isBlocked());  
+ if (focusDescendant != null) {  
+ if (isAncestor(focusDescendant)) {  
 focusDescendant.requestFocus(true);  
========================================

Various fixes to ListSelection.  
  
Project: pivot.txt  
Commit: 0ef49561cfd272cd474f3178f6cb72285fa6fa63  
ClassName: org.apache.pivot.wtk.ListSelection.java  
  
 public Sequence<Span> removeRange(int start, int end) {  
**- assert(start > 0);  
+ assert(start >= 0);  
 assert(end >= start);**========================================

fix some eclipse warnings  
  
Project: pivot.txt  
Commit: 222f70a0b4306f10ba405eb53467e7ec90bd85a0  
ClassName: org.apache.pivot.wtk.skin.terra.TerraSpinnerSkin.java  
  
- public void start(Spinner spinner, int direction) {  
**- assert(direction != 0) : "Direction must be positive or negative";**+ public void start(Spinner spinnerArgument, int directionArgument) {  
**+ assert(directionArgument != 0) : "Direction must be positive or negative";**========================================

Minor fixes to TextAreaSkin and TerraFormSkin.  
  
Project: pivot.txt  
Commit: 11e0c2d3a98a0eb18f842ea22e9c71363593ec65  
ClassName: org.apache.pivot.wtk.skin.TextAreaSkin.java  
  
 public void setBreakWidth(int breakWidth) {  
**- assert (breakWidth < 0);  
+ assert (breakWidth > 0);**========================================

Minor resize bug fix.  
  
Project: pivot.txt  
Commit: 13bd743a797824d485353e48d506b27a7522b0c9  
ClassName: pivot.wtk.skin.terra.AbstractFrameSkin.java  
  
 public void mouseUp(Component component, Mouse.Button button, int x, int y) {  
- Display display = component.getWindow().getDisplay();  
- display.getComponentMouseListeners().remove(this);  
- display.getComponentMouseButtonListeners().remove(this);  
**+ assert (component instanceof Display);**+ component.getComponentMouseListeners().remove(this);  
+ component.getComponentMouseButtonListeners().remove(this);  
 }  
========================================

Trust13 related fixes  
  
Project: rampart.txt  
Commit: 9577b6ea7bc364060edb8de1fcff75c1fb6763ed  
ClassName: org.apache.rahas.client.STSClient.java  
  
@@ -555,3 +562,3 @@ private OMElement createIssueRequest(String requestType,  
**- log.debug("Processing Trust10 assertion");  
+ log.debug("Processing Trust assertion");**========================================

validation binding issues  
  
Project: rampart.txt  
Commit: 2c311601acb577e07f301b6ac0896c6ab72f33d0  
ClassName: org.apache.rahas.impl.SAMLTokenValidator.java  
  
+ private boolean isValid(Token token, PublicKey issuerPBKey) {  
**+ OMElement assertionOMElement = token.getToken();  
+ SAMLAssertion samlAssertion = null;**+ try {  
**+ samlAssertion = new SAMLAssertion((Element) assertionOMElement);**+ log.info("Verifying token validity...");  
**+ samlAssertion.verify(issuerPBKey);**+ } catch (SAMLException e) {  
+ log.error("Could not verify signature", e);  
+ return false;  
+ }  
+ return true;  
+ }  
========================================

Fixing https://issues.apache.org/jira/browse/RAMPART-286  
  
Project: rampart.txt  
Commit: ba4dee432a673d815bb546c3e76947ca7eac293f  
ClassName: org.apache.ws.secpolicy12.builders.AsymmetricBindingBuilder.java  
  
**@@ -80,2 +80,3 @@ private void processAlternative(List assertions, AsymmetricBinding asymmetricBin** asymmetricBinding.setIncludeTimestamp(true);  
**+ asymmetricBinding.setIncludeTimestampOptional(assertion.isOptional());**========================================

Fixing https://issues.apache.org/jira/browse/RAMPART-286  
  
Project: rampart.txt  
Commit: ba4dee432a673d815bb546c3e76947ca7eac293f  
ClassName: org.apache.ws.secpolicy11.builders.AsymmetricBindingBuilder.java  
  
**@@ -80,2 +80,3 @@ private void processAlternative(List assertions, AsymmetricBinding asymmetricBin** asymmetricBinding.setIncludeTimestamp(true);  
**+ asymmetricBinding.setIncludeTimestampOptional(assertion.isOptional());**========================================

Fixing https://issues.apache.org/jira/browse/RAMPART-285 - Thanks Thilina for the patch  
  
Project: rampart.txt  
Commit: 961bbbecbe576288a39cd06e560cedf955467a82  
ClassName: org.apache.rahas.impl.SAML2TokenIssuer.java  
  
@@ -193,2 +190,7 @@ public SOAPEnvelope issue(RahasData data) throws TrustException {  
+ Conditions conditions = new ConditionsBuilder().buildObject();  
+ conditions.setNotBefore(creationDate);  
+ conditions.setNotOnOrAfter(expirationDate);  
**+ assertion.setConditions(conditions);**========================================

Applying patch for Rampart-42 bug. Adding the new classes  
  
Project: rampart.txt  
Commit: 8936c6b31464d4cbc621ecf72a850b2662aa1c9a  
ClassName: org.apache.rampart.policy.builders.SSLConfigBuilder.java  
  
**+ public Assertion build(OMElement element, AssertionBuilderFactory factory)**+ throws IllegalArgumentException {  
+ SSLConfig sslCofig = new SSLConfig();   
+ Properties properties = new Properties();  
+ OMElement childElement;  
+ OMAttribute name;  
+ String value;   
+ for (Iterator iterator = element.getChildElements(); iterator.hasNext();) {   
+ childElement = (OMElement) iterator.next();  
+ QName prop = new QName(RampartConfig.NS, SSLConfig.PROPERTY\_LN);  
+ if (prop.equals(childElement.getQName())) {  
+ name = childElement.getAttribute(new QName(SSLConfig.PROPERTY\_NAME\_ATTR));  
+ value = childElement.getText();  
+ System.setProperty(name.getAttributeValue(), value);  
+ properties.put(name.getAttributeValue(), value.trim());  
+ }  
+ }   
+ sslCofig.setProp(properties);  
+ return sslCofig;  
+ }  
========================================

Fixing https://issues.apache.org/jira/browse/RAMPART-285 - Thanks Thilina for the patch  
  
Project: rampart.txt  
Commit: 961bbbecbe576288a39cd06e560cedf955467a82  
ClassName: org.apache.rampart.RampartEngine.java  
  
@@ -184,6 +185,25 @@ public Vector process(MessageContext msgCtx) throws WSSPolicyException,  
 **Subject subject = assertion.getSubject();**+ Date dateOfCreation = null;  
+ Date dateOfExpiration = null;  
**+ if (assertion.getConditions() != null) {  
+ Conditions conditions = assertion.getConditions();**+ if (conditions.getNotBefore() != null) {  
+ dateOfCreation = conditions.getNotBefore().toDate();  
+ }  
+ if (conditions.getNotOnOrAfter() != null) {  
+ dateOfExpiration = conditions.getNotOnOrAfter().toDate();  
+ }  
+ } else {  
 SubjectConfirmationData scData = subject.getSubjectConfirmations()  
 .get(0).getSubjectConfirmationData();  
- Date dateOfCreation = scData.getNotBefore().toDate();  
- Date dateOfExpiration = scData.getNotOnOrAfter().toDate();  
+ if (scData.getNotBefore() != null) {  
+ dateOfCreation = scData.getNotBefore().toDate();  
+ }  
+ if (scData.getNotOnOrAfter() != null) {  
+ dateOfExpiration = scData.getNotOnOrAfter().toDate();  
+ }  
+ }  
========================================

Fixing issue http://issues.apache.org/jira/browse/RAMPART-140.  
  
Project: rampart.txt  
Commit: 41cfd8103b94a0b5044d0eda67801e5ff7ec1788  
ClassName: org.apache.rampart.handler.PostDispatchVerificationHandler.java  
  
@@ -123,4 +126,26 @@ public InvocationResponse invoke(MessageContext msgContext)  
 securityPolicyPresent = true;  
+ break;  
**+ } else if (assertion instanceof SupportingToken) {**========================================

Trust13 related fixes  
  
Project: rampart.txt  
Commit: 9577b6ea7bc364060edb8de1fcff75c1fb6763ed  
ClassName: org.apache.rahas.client.STSClient.java  
  
@@ -499,2 +502,6 @@ private void processPolicy(Policy issuerPolicy, Policy servicePolicy) {  
 **this.trust10 = (Trust10) tempAssertion;  
+ } else if (tempAssertion instanceof Trust13) {  
+ log.debug("Extracting Trust13 assertion from " +**+ "service policy");  
**+ this.trust13 = (Trust13) tempAssertion;** }  
========================================

**Fixed the problem of org.apache.rahas.client.STSClient.org.apache.rahas.processIssueResponse failing if SamlAssertion is issued - RAMPART-154**  
Project: rampart.txt  
Commit: a09f86965d59dda1c8662a4da85ff8e6edda3af9  
ClassName: org.apache.rahas.client.STSClient.java  
  
@@ -549,2 +549,7 @@ private String findIdentifier(OMElement reqAttRef,  
 id = token.getAttributeValue(new QName(WSConstants.WSU\_NS, "Id"));  
+ if ( id == null )  
+ {  
**+ id = token.getAttributeValue(new QName( "AssertionID"));**+ }  
========================================

global rename Activity --> Task. Java part compiles and runs unit tests; typos not fixed yet (e.g. "an task" in the comments)  
  
Project: reef.txt  
Commit: 3f4519cf5a7cac68ab4af743e8a4f0e3b7b24604  
ClassName: com.microsoft.reef.runtime.common.driver.EvaluatorManager.java  
  
- private void handle(final ReefServiceProtos.ActivityStatusProto activityStatusProto) {  
+ private void handle(final ReefServiceProtos.TaskStatusProto taskStatusProto) {  
- LOG.log(Level.FINEST, "Received activity {0} status {1}",  
- new Object[] { activityStatusProto.getActivityId(), activityStatusProto.getState() });  
+ LOG.log(Level.FINEST, "Received task {0} status {1}",  
+ new Object[]{taskStatusProto.getTaskId(), taskStatusProto.getState()});  
- final String activityId = activityStatusProto.getActivityId();  
- final String contextId = activityStatusProto.getContextId();  
- final ReefServiceProtos.State activityState = activityStatusProto.getState();  
+ final String taskId = taskStatusProto.getTaskId();  
+ final String contextId = taskStatusProto.getContextId();  
+ final ReefServiceProtos.State taskState = taskStatusProto.getState();  
- if (ReefServiceProtos.State.INIT == activityState) {  
+ if (ReefServiceProtos.State.INIT == taskState) {  
 final EvaluatorContext evaluatorContext = getEvaluatorContext(contextId);  
- this.runningActivity = new RunningActivityImpl(this, activityId, evaluatorContext);  
- this.dispatcher.onNext(RunningActivity.class, this.runningActivity);  
- } else if (ReefServiceProtos.State.SUSPEND == activityState) {  
+ this.runningTask = new RunningTaskImpl(this, taskId, evaluatorContext);  
+ this.dispatcher.onNext(RunningTask.class, this.runningTask);  
+ } else if (ReefServiceProtos.State.SUSPEND == taskState) {  
 final EvaluatorContext evaluatorContext = getEvaluatorContext(contextId);  
- this.runningActivity = null;  
- final byte[] message = activityStatusProto.hasResult() ? activityStatusProto.getResult().toByteArray() : null;  
- this.dispatcher.onNext(SuspendedActivity.class,  
- new SuspendedActivityImpl(evaluatorContext, message, activityId));  
- } else if (ReefServiceProtos.State.DONE == activityState) {  
+ this.runningTask = null;  
+ final byte[] message = taskStatusProto.hasResult() ? taskStatusProto.getResult().toByteArray() : null;  
+ this.dispatcher.onNext(SuspendedTask.class,  
+ new SuspendedTaskImpl(evaluatorContext, message, taskId));  
+ } else if (ReefServiceProtos.State.DONE == taskState) {  
 final EvaluatorContext evaluatorContext = getEvaluatorContext(contextId);  
- this.runningActivity = null;  
- final byte[] message = activityStatusProto.hasResult() ? activityStatusProto.getResult().toByteArray() : null;  
- this.dispatcher.onNext(CompletedActivity.class,  
- new CompletedActivityImpl(evaluatorContext, message, activityId));  
- } else if (ReefServiceProtos.State.FAILED == activityState) {  
- this.runningActivity = null;  
+ this.runningTask = null;  
+ final byte[] message = taskStatusProto.hasResult() ? taskStatusProto.getResult().toByteArray() : null;  
+ this.dispatcher.onNext(CompletedTask.class,  
+ new CompletedTaskImpl(evaluatorContext, message, taskId));  
+ } else if (ReefServiceProtos.State.FAILED == taskState) {  
+ this.runningTask = null;  
 final ObjectSerializableCodec<Exception> codec = new ObjectSerializableCodec<>();  
 final EvaluatorContext evaluatorContext = getEvaluatorContext(contextId);  
- final FailedActivity activityException = activityStatusProto.hasResult() ?  
- new FailedActivity(activityId, codec.decode(activityStatusProto.getResult().toByteArray()), Optional.<ActiveContext>of(evaluatorContext)) :  
- new FailedActivity(activityId, "Failed Activity: " + activityState, Optional.<ActiveContext>of(evaluatorContext));  
+ final FailedTask taskException = taskStatusProto.hasResult() ?  
+ new FailedTask(taskId, codec.decode(taskStatusProto.getResult().toByteArray()), Optional.<ActiveContext>of(evaluatorContext)) :  
+ new FailedTask(taskId, "Failed Task: " + taskState, Optional.<ActiveContext>of(evaluatorContext));  
- this.dispatcher.onNext(FailedActivity.class, activityException);  
- } else if (activityStatusProto.getActivityMessageCount() > 0) {  
**- assert (this.runningActivity != null);**- for (final ReefServiceProtos.ActivityStatusProto.ActivityMessageProto activityMessageProto : activityStatusProto.getActivityMessageList()) {  
- this.dispatcher.onNext(ActivityMessage.class,  
- new ActivityMessageImpl(activityMessageProto.getMessage().toByteArray(),  
- activityId, contextId, activityMessageProto.getSourceId()));  
+ this.dispatcher.onNext(FailedTask.class, taskException);  
+ } else if (taskStatusProto.getTaskMessageCount() > 0) {  
**+ assert (this.runningTask != null);**+ for (final ReefServiceProtos.TaskStatusProto.TaskMessageProto taskMessageProto : taskStatusProto.getTaskMessageList()) {  
+ this.dispatcher.onNext(TaskMessage.class,  
+ new TaskMessageImpl(taskMessageProto.getMessage().toByteArray(),  
+ taskId, contextId, taskMessageProto.getSourceId()));  
 }  
========================================

global rename Activity --> Task. Java part compiles and runs unit tests; typos not fixed yet (e.g. "an task" in the comments)  
  
Project: reef.txt  
Commit: 3f4519cf5a7cac68ab4af743e8a4f0e3b7b24604  
ClassName: com.microsoft.reef.task.TaskMessage.java  
  
+ public static TaskMessage from(final String messageSourceID, final byte[] theBytes) {  
**+ assert (theBytes != null && messageSourceID != null);**+ return new TaskMessage(messageSourceID, theBytes);  
+ }  
========================================

[REEF-30] Adding reef-runtime-mesos  
This adds reef-runtime-mesos. Please refer to the pull request for details:  
https://github.com/apache/incubator-reef/pull/52  
JIRA: [REEF-30]: https://issues.apache.org/jira/browse/REEF-30  
Pull Request: This closes #52  
  
Project: reef.txt  
Commit: c908a526d8ad14ab9a35be351a89160fe6576992  
ClassName: org.apache.reef.runtime.mesos.evaluator.REEFExecutor.java  
  
+ public final void onEvaluatorLaunch(final EvaluatorLaunch evaluatorLaunch) {  
+ LOG.log(Level.INFO, "Launch!!!! {0}", evaluatorLaunch.toString());  
**+ assert(evaluatorLaunch.getIdentifier().toString().equals(this.mesosExecutorId));**+ final ExecutorService evaluatorLaunchExecutorService = Executors.newSingleThreadExecutor();  
+ evaluatorLaunchExecutorService.submit(new Thread() {  
+ public void run() {  
+ try {  
+ final List<String> command = Arrays.asList(evaluatorLaunch.getCommand().toString().split(" "));  
+ LOG.log(Level.INFO, "Command!!!! {0}", command);  
+ final FileSystem fileSystem = FileSystem.get(new Configuration());  
+ final Path hdfsFolder = new Path(fileSystem.getUri() + "/" + mesosExecutorId);  
+ final File localFolder = new File(fileNames.getREEFFolderName(), fileNames.getLocalFolderName());  
+ FileUtil.copy(fileSystem, hdfsFolder, localFolder, true, new Configuration());  
+ evaluatorProcess = new ProcessBuilder()  
+ .command(command)  
+ .redirectError(new File(fileNames.getEvaluatorStderrFileName()))  
+ .redirectOutput(new File(fileNames.getEvaluatorStdoutFileName()))  
+ .start();  
+ evaluatorProcessExitValue = evaluatorProcess.waitFor();  
+ fileSystem.close();  
+ } catch (IOException | InterruptedException e) {  
+ throw new RuntimeException(e);  
+ }  
+ }  
+ });  
+ evaluatorLaunchExecutorService.shutdown();  
+ }  
========================================

[REEF-30] Adding reef-runtime-mesos  
This adds reef-runtime-mesos. Please refer to the pull request for details:  
https://github.com/apache/incubator-reef/pull/52  
JIRA: [REEF-30]: https://issues.apache.org/jira/browse/REEF-30  
Pull Request: This closes #52  
  
Project: reef.txt  
Commit: c908a526d8ad14ab9a35be351a89160fe6576992  
ClassName: org.apache.reef.runtime.mesos.evaluator.REEFExecutor.java  
  
+ public final void onEvaluatorRelease(final EvaluatorRelease evaluatorRelease) {  
+ LOG.log(Level.INFO, "Release!!!! {0}", evaluatorRelease.toString());  
**+ assert(evaluatorRelease.getIdentifier().toString().equals(this.mesosExecutorId));**+ this.onStop();  
+ }  
========================================

Cleaning up of the Driver idle detection  
Closes #934  
This PR restructures the Driver's idle handling. It adds `DriverIdleManager` that joins all the different sources for potential idleness. While the set of sources is extensible, it right now contains three:  
\* The clock  
\* The resource manager  
\* The application, represented by its event handlers.  
Each of these trigger a check for idleness when they are idle. Also, they are all considered when another component triggers this check. All of this is logged.  
\*Open Issues:\* the unit test for #906 introduced in #985 still doesn't pass. This is because we don't separate the resource manager from the REEF threads. This forces the `FailedEvaluator` in that test to come in \*after\* idle is detected. I created #986 to track this issue.  
  
Project: reef.txt  
Commit: 685e4ac75466ce08035a4c309e2ebd631f94ed5a  
ClassName: com.microsoft.reef.runtime.common.driver.resourcemanager.ResourceManagerStatus.java  
  
+ private synchronized void onRMRunning(final DriverRuntimeProtocol.RuntimeStatusProto runtimeStatusProto) {  
**+ assert (runtimeStatusProto.getState() == ReefServiceProtos.State.RUNNING);**+ if (this.isIdle()) {  
+ this.driverIdleManager.get().onPotentiallyIdle(IDLE\_MESSAGE);  
+ }  
+ }  
========================================

[REEF-3]: A basic Task Scheduler  
This implements REEF-3: https://issues.apache.org/jira/browse/REEF-3  
It is a basic Task Scheduler example using Reef-webserver. The application  
receives the task (shell command) list from user and execute the tasks in a  
FIFO order.  
Users can send the HTTP request to the server via URL:  
http://{address}:{port}/reef-example-scheduler/v1  
Users can send the following requests  
\* /list lists all the tasks' status.  
\* /clear clears all the tasks waiting on the queue. Returns how  
many tasks are removed.  
\* /submit?cmd=COMMAND submits a COMMAND to be executed and return the taskID.  
\* /status?id=ID returns the status of task whose id is ID.  
\* /cancel?id=ID cancels the task whose id is ID.  
The results of tasks are written in the log files - both in driver's and  
evaluators'.  
By default, the application reuses existing evaluator to run Tasks.  
Using -retain false option, the driver allocates an evaluator for each command.  
The code was contributed by Yunseong Lee <yunseong.lee@me.com>  
Closes #3  
  
Project: reef.txt  
Commit: 10f96514379179b8b2cf11fcf350418787302078  
ClassName: org.apache.reef.examples.scheduler.SchedulerDriver.java  
  
+ public void onNext(final AllocatedEvaluator evaluator) {  
+ LOG.log(Level.INFO, "Evaluator is ready");  
**+ assert (state == State.WAIT\_EVALUATOR);**+ evaluator.submitContext(ContextConfiguration.CONF  
+ .set(ContextConfiguration.IDENTIFIER, "SchedulerContext")  
+ .build());  
+ }  
========================================

separated GraphVisitor into NodeVisitor and EdgeVisitor interfaces + minor related fixes in Walk and around.  
  
Project: reef.txt  
Commit: fcebc342628246809f251cfe9e1c2803e2a867d0  
ClassName: com.microsoft.tang.util.walk.Walk.java  
  
+ public static boolean preorder(  
+ final NodeVisitor aNodeVisitor, final EdgeVisitor aEdgeVisitor, final Configuration aConfig)  
+ {  
**+ assert (aNodeVisitor != null || aEdgeVisitor != null);**+ final Node root = aConfig.getClassHierarchy().getNamespace();  
+ return preorder(aNodeVisitor, aEdgeVisitor, root);  
+ }  
========================================

[REEF-3]: A basic Task Scheduler  
This implements REEF-3: https://issues.apache.org/jira/browse/REEF-3  
It is a basic Task Scheduler example using Reef-webserver. The application  
receives the task (shell command) list from user and execute the tasks in a  
FIFO order.  
Users can send the HTTP request to the server via URL:  
http://{address}:{port}/reef-example-scheduler/v1  
Users can send the following requests  
\* /list lists all the tasks' status.  
\* /clear clears all the tasks waiting on the queue. Returns how  
many tasks are removed.  
\* /submit?cmd=COMMAND submits a COMMAND to be executed and return the taskID.  
\* /status?id=ID returns the status of task whose id is ID.  
\* /cancel?id=ID cancels the task whose id is ID.  
The results of tasks are written in the log files - both in driver's and  
evaluators'.  
By default, the application reuses existing evaluator to run Tasks.  
Using -retain false option, the driver allocates an evaluator for each command.  
The code was contributed by Yunseong Lee <yunseong.lee@me.com>  
Closes #3  
  
Project: reef.txt  
Commit: 10f96514379179b8b2cf11fcf350418787302078  
ClassName: org.apache.reef.examples.scheduler.SchedulerDriver.java  
  
+ public void onNext(ActiveContext context) {  
+ synchronized (lock) {  
+ LOG.log(Level.INFO, "Context available : {0}", context.getId());  
**+ assert (state == State.WAIT\_EVALUATOR);**+ state = State.READY;  
+ waitForCommands(context);  
+ }  
+ }  
========================================

Cleaning up of the Driver idle detection  
Closes #934  
This PR restructures the Driver's idle handling. It adds `DriverIdleManager` that joins all the different sources for potential idleness. While the set of sources is extensible, it right now contains three:  
\* The clock  
\* The resource manager  
\* The application, represented by its event handlers.  
Each of these trigger a check for idleness when they are idle. Also, they are all considered when another component triggers this check. All of this is logged.  
\*Open Issues:\* the unit test for #906 introduced in #985 still doesn't pass. This is because we don't separate the resource manager from the REEF threads. This forces the `FailedEvaluator` in that test to come in \*after\* idle is detected. I created #986 to track this issue.  
  
Project: reef.txt  
Commit: 685e4ac75466ce08035a4c309e2ebd631f94ed5a  
ClassName: com.microsoft.reef.runtime.common.driver.resourcemanager.ResourceManagerStatus.java  
  
+ private synchronized void onRMFailure(final DriverRuntimeProtocol.RuntimeStatusProto runtimeStatusProto) {  
**+ assert (runtimeStatusProto.getState() == ReefServiceProtos.State.FAILED);**+ this.resourceManagerErrorHandler.onNext(runtimeStatusProto.getError());  
+ }  
========================================

Cleaning up of the Driver idle detection  
Closes #934  
This PR restructures the Driver's idle handling. It adds `DriverIdleManager` that joins all the different sources for potential idleness. While the set of sources is extensible, it right now contains three:  
\* The clock  
\* The resource manager  
\* The application, represented by its event handlers.  
Each of these trigger a check for idleness when they are idle. Also, they are all considered when another component triggers this check. All of this is logged.  
\*Open Issues:\* the unit test for #906 introduced in #985 still doesn't pass. This is because we don't separate the resource manager from the REEF threads. This forces the `FailedEvaluator` in that test to come in \*after\* idle is detected. I created #986 to track this issue.  
  
Project: reef.txt  
Commit: 685e4ac75466ce08035a4c309e2ebd631f94ed5a  
ClassName: com.microsoft.reef.runtime.common.driver.resourcemanager.ResourceManagerStatus.java  
  
+ private synchronized void onRMDone(final DriverRuntimeProtocol.RuntimeStatusProto runtimeStatusProto) {  
**+ assert (runtimeStatusProto.getState() == ReefServiceProtos.State.DONE);**+ LOG.log(Level.INFO, "Resource Manager shutdown happened. Triggering Driver shutdown.");  
+ this.driverStatusManager.onComplete();  
+ }  
========================================

Added code for interop scenario 4.1 and fixed a few token reference issues in RampartBasedSecurityManager.  
  
Project: sandesha.txt  
Commit: ecd88ffd1e6a62bf4f9946a90b3badec4c2e986c  
ClassName: org.apache.sandesha2.security.rampart.RampartBasedSecurityManager.java  
  
@@ -200,4 +220,8 @@ public SecurityToken getSecurityToken(MessageContext message)  
 Policy issuerPolicy = secConvTok.getBootstrapPolicy();  
**+ issuerPolicy.addAssertion(rpd.getRampartConfig());** STSClient client = new STSClient(message.getConfigurationContext());  
+ Options op = new Options();  
+ op.setProperty(SandeshaClientConstants.UNRELIABLE\_MESSAGE, Constants.VALUE\_TRUE);  
+ client.setOptions(op);  
 client.setAction(action);  
========================================

Apply fixes for SANDESHA2-147 and SANDESHA2-150  
  
Project: sandesha.txt  
Commit: e9dd897b2e8f124c66f6729175a75a0cd96fef0c  
ClassName: org.apache.sandesha2.util.SandeshaUtil.java  
  
**@@ -465,3 +465,3 @@ public static void assertProofOfPossession(RMSequenceBean bean, MessageContext c** if (log.isDebugEnabled())   
**- log.debug("Enter: SandeshaUtil::assertProofOfPossession :" + bean + ", " + context + ", " + elementToCheck);  
+ log.debug("Enter: SandeshaUtil::assertProofOfPossession :" + bean + ", " + context + ", " + elementToCheck.getQName());**========================================

###############################################  
IMPORTANT:  
- The different classes do not call Init.init() any longer. This must be done  
by YOU in your application. If you miss that, you'll get many  
AlgorithmNotRegistered exceptions...  
###############################################  
Summary:  
- The software is faster. Especially canonicalization is between  
factor 5--80 faster than the old one.  
- Some deprecated methods in the Canonicalizer are deleted.  
- We support Exclusive Canonicalization  
- We support the XPath Filter version 2.0 Draft.  
###############################################  
Optimizations and speed-up  
- canonicalization  
- inclusive c14n is now faster (factor between 5 and 80)  
- transforms  
- enveloped-signature is now faster (no XPath ops any more)  
- base64 is now faster (no XPath ops any more)  
- c14n is now faster (due to faster c14n algo)  
###############################################  
Signature package:  
- The XMLSignatureInput which is used for passing node sets and octet  
streams into transforms and which is also the result of transforms  
uses a java.util.Set now instead of a NodeList for the internal  
representation of xpath node sets. This allows easier queries in the  
form: Is node N part of the node set.  
The implication is that you can also pass a Set which contains the nodes  
to be canonicalized to the Canonicalizers using  
public byte[] canonicalizeXPathNodeSet(Set xpathNodeSet)  
###############################################  
Canonicalizer:  
- A bug (well, my understanding of c14n) is corrected regarding the  
canonicalization of node sets. That bug related to the xml:\*  
attributes. See xmldsig mailing list archive @ w3.org for details.  
- removed are the methods  
- public byte[] canonicalize(Node node)  
- public byte[] canonicalizeDocument(Document doc)  
- public byte[] canonicalizeSingleNode(Node rootNode)  
replaced by public byte[] canonicalizeSubtree(Node node)  
- public byte[] canonicalize(NodeList xpathNodeSet)  
replaced by public byte[] canonicalizeXPathNodeSet(NodeList xpathNodeSet)  
- public void setXPath(Object xpath)  
- public Object getXPath()  
- public String getXPathString()  
- public void setXPathNodeSet(NodeList nodeList)  
These are no longer in use. If you want to c14nize an xpath  
node set, select it using CachedXPathAPI and then apply  
canonicalizeXPathNodeSet to the node set.  
- public void setRemoveNSAttrs(boolean remove)  
- public boolean getRemoveNSAttrs()  
The c14nizers do not add any attributes (namespaces or xml:\*)  
to the document, so these method make no sense.  
- The Canonicalizer now supports "Exclusive XML Canonicalization  
Version 1.0" <http://www.w3.org/Signature/Drafts/xml-exc-c14n>, Rev 1.58.  
For that reason, the c14n methods allow an additional String parameter  
for passing the inclusive namespaces.  
public byte[] canonicalizeSubtree(Node node,  
String inclusiveNamespaces)  
public byte[] canonicalizeXPathNodeSet(NodeList xpathNodeSet,  
String inclusiveNamespaces)  
Such a string looks e.g. like this  
String inclusiveNamespaces = "ds xenc ex #default";  
For more on exclusive c14n, see the spec. If you pass this parameter to the  
regular (inclusive) c14nizer, you'll get a  
CanonicalizationException("c14n.Canonicalizer.UnsupportedOperation")  
###############################################  
Transforms:  
- The exclusive c14n is also supported by the transform framework.  
The parameter for the inclusive namespaces is the class  
org.apache.xml.security.transforms.params.InclusiveNamespaces  
If you want to make a Transform like this, do that:  
Document doc = ...;  
Transforms transforms = new Transforms(doc);  
InclusiveNamespaces incNS = new InclusiveNamespaces(doc, "ns2");  
transforms.addTransform(Transforms.TRANSFORM\_C14N\_EXCL\_OMIT\_COMMENTS,  
incNS.getElement());  
- The XPathContainer for the XPath transform is now moved from the  
org.apache.xml.security.c14n.helper package to  
org.apache.xml.security.transforms.params.XPathContainer  
- The enveloped-signature transform is faster now. We don't do costly  
XPath operations but 'simple' DOM ops.  
- Base64 is faster (no XPath ops).  
- The TransformXPath2Filter is now supported by the package. It can be used by  
using the identifier Transforms.TRANSFORM\_XPATH2FILTER in conjuction with the  
XPath2FilterContainer for passing parameters. To know what xfilter2 is, see  
http://www.w3.org/Signature/Drafts/xmldsig-xfilter2/ :  
Document doc = ...;  
Transforms transforms = new Transforms(doc);  
XPath2FilterContainer x2c =  
// intersect  
XPath2FilterContainer.newInstanceIntersect(doc, "//a");  
// subtract  
XPath2FilterContainer.newInstanceSubtract(doc, "//a");  
// union  
XPath2FilterContainer.newInstanceUnion(doc, "//a");  
transforms.addTransform(Transforms.TRANSFORM\_XPATH2FILTER,  
x2c.getElement());  
  
Project: santuario-java.txt  
Commit: 237586bb9f5dd493f2eddbb31826df56d21dce4c  
ClassName: org.apache.xml.security.c14n.helper.C14nHelper.java  
  
+ public static void checkForRelativeNamespace(Element ctxNode)  
+ throws CanonicalizationException {  
+ if (ctxNode != null) {  
+ NamedNodeMap attributes = ctxNode.getAttributes();  
+ for (int i = 0; i < attributes.getLength(); i++) {  
**+ C14nHelper.assertNotRelativeNS((Attr) attributes.item(i));**+ }  
+ } else {  
+ throw new CanonicalizationException(  
+ "Called checkForRelativeNamespace() on null");  
+ }  
 }  
========================================

Fixed bug 39286: Add support for SAML ID attribute type recognition to  
IdResolver. Formatted code with consistent indents.  
  
Project: santuario-java.txt  
Commit: e760ba7b5509f232f023452facf21be06e68e217  
ClassName: org.apache.xml.security.utils.IdResolver.java  
  
@@ -173,3 +165,4 @@ private static Element getElementByIdType(Document doc, String id) {  
 "http://schemas.xmlsoap.org/soap/security/2000-12",  
- "http://www.w3.org/2002/03/xkms#"   
+ "http://www.w3.org/2002/03/xkms#",  
**+ "urn:oasis:names:tc:SAML:1.0:assertion"** };  
========================================

Fix timing issues with ExchangeCompletedListener  
  
Project: servicemix3.txt  
Commit: 8c86e23053fdad99c3b84a9edd3c0504cd84bd6e  
ClassName: org.apache.servicemix.tck.ExchangeCompletedListener.java  
  
 **public void assertExchangeCompleted() throws Exception {**- Thread.sleep(50);  
+ long start = System.currentTimeMillis();  
+ MessageExchange active = null;  
+ while (true) {  
+ synchronized (exchanges) {  
 for (Iterator it = exchanges.values().iterator(); it.hasNext();) {  
+ active = null;  
 MessageExchange me = (MessageExchange) it.next();  
**- assertTrue("Exchange is ACTIVE", me.getStatus() != ExchangeStatus.ACTIVE);**+ if (me.getStatus() == ExchangeStatus.ACTIVE) {  
+ active = me;  
+ break;  
+ }  
+ }  
+ if (active == null) {  
+ break;  
+ }  
+ long remain = timeout - (System.currentTimeMillis() - start);  
+ if (remain < 0) {  
**+ assertTrue("Exchange is ACTIVE", active.getStatus() != ExchangeStatus.ACTIVE);**+ } else {  
+ exchanges.wait(remain);  
+ }  
+ }  
 }  
========================================

SMX4KNL-104: When using a remote connection, errors are not reported correctly  
  
Project: servicemix4-kernel.txt  
Commit: 2ed67ca2fbb7f87f66b5dfcfa6ea6a8a28504f16  
ClassName: org.apache.geronimo.gshell.remote.client.proxy.SpringRemoteShellProxy.java  
  
+ private void displayError(final Throwable error) {  
**+ assert error != null;**+ Throwable cause = error;  
+ if (error instanceof ErrorNotification) {  
+ cause = error.getCause();  
+ }  
+ io.err.print("@|bold,red ERROR| ");  
+ io.err.print(cause.getClass().getSimpleName());  
+ io.err.println(": @|bold,red " + cause.getMessage() + "|");  
+ String stackTraceProperty = System.getProperty("gshell.show.stacktrace");  
+ boolean stackTraceFlag = false;  
+ if (stackTraceProperty != null) {  
+ stackTraceFlag = stackTraceProperty.trim().equals("true");  
+ }  
+ if (io.isDebug()) {  
+ log.debug(error.toString(), error);  
+ }  
+ else if (io.isVerbose() || stackTraceFlag) {  
+ StackTraceElement[] trace = cause.getStackTrace();  
+ StringBuffer buff = new StringBuffer();  
+ for (StackTraceElement e : trace) {  
+ buff.append(" @|bold at| ").  
+ append(e.getClassName()).  
  
+ append(".").  
+ append(e.getMethodName()).  
+ append(" (@|bold ");  
+ buff.append(e.isNativeMethod() ? "Native Method" :  
+ (e.getFileName() != null && e.getLineNumber() != -1 ? e.getFileName() + ":" + e.getLineNumber() :  
+ (e.getFileName() != null ? e.getFileName() : "Unknown Source")));  
+ buff.append("|)");  
+ io.err.println(buff);  
+ buff.setLength(0);  
+ }  
+ }  
+ }  
========================================

SMX4KNL-104: When using a remote connection, errors are not reported correctly  
  
Project: servicemix4-kernel.txt  
Commit: 2ed67ca2fbb7f87f66b5dfcfa6ea6a8a28504f16  
ClassName: org.apache.geronimo.gshell.remote.client.proxy.SpringRemoteShellProxy.java  
  
+ public void run(final Object... args) throws Exception {  
**+ assert args != null;**+ ensureOpened();  
+ log.debug("Starting interactive console; args: {}", args);  
+ final AtomicReference<ExitNotification> exitNotifHolder = new AtomicReference<ExitNotification>();  
+ final AtomicReference<Object> lastResultHolder = new AtomicReference<Object>();  
+ Console.Executor executor = new Console.Executor() {  
+ public Result execute(final String line) throws Exception {  
**+ assert line != null;**+ try {  
+ Object result = SpringRemoteShellProxy.this.execute(line);  
+ lastResultHolder.set(result);  
+ }  
+ catch (ExitNotification n) {  
+ exitNotifHolder.set(n);  
+ return Result.STOP;  
+ }  
+ return Result.CONTINUE;  
+ }  
+ };  
+ JLineConsole console = new JLineConsole(executor, io, terminal);  
+ console.setPrompter(new Console.Prompter() {  
+ Renderer renderer = new Renderer();  
+ public String prompt() {  
+ String userName = "user"; // shellInfo.getUserName();  
+ String hostName = "remote"; // shellInfo.getLocalHost().getHostName();  
+ String path = "/";  
+ return renderer.render("@|bold " + userName + "|@" + hostName + ":@|bold " + path + "|> ");  
+ }  
+ });  
+ console.setErrorHandler(new Console.ErrorHandler() {  
+ public Result handleError(final Throwable error) {  
**+ assert error != null;**+ displayError(error);  
+ return Result.CONTINUE;  
+ }  
+ });  
+ History history = new History();  
+ console.setHistory(history);  
+ console.setHistoryFile(new File(branding.getUserDirectory(), branding.getHistoryFileName()));  
+ if (!io.isQuiet()) {  
+ io.out.println(branding.getWelcomeBanner());  
+ }  
+ if (args.length != 0) {  
+ execute(args);  
+ }  
+ console.run();  
+ ExitNotification n = exitNotifHolder.get();  
+ if (n != null) {  
+ throw n;  
+ }  
+ }  
========================================

Continued work homogenizing Session access - previous work on DefaultSubjectFactory and DefaultSecurityManager allowed the removal of the WebSubjectFactory entirely. Made minor code enhancements per IDEA inspections. Enabled log4j as the logging mechanism for test cases, but will probably change that to be Logback so we can have one less jar dependency. Lots of minor JavaDoc fixes  
  
Project: shiro.txt  
Commit: 9a2251bb03c6ccbf610fb8a1933bcf081d7b1548  
ClassName: org.apache.shiro.session.mgt.AbstractValidatingSessionManager.java  
  
**- private static void assertNotNull(Session session, Serializable sessionId) throws UnknownSessionException {  
+ private void assertNotNull(Session session, Serializable sessionId) throws UnknownSessionException {** if (session == null) {  
========================================

Additional name change fixes related to jsecurity/ki -> shiro migration (changed case-sensitive references). Re-enabled the spring-hibernate module, cleaned up dependency management by excluding all uses of commons-logging and using jcl-over-slf4j where requried by 3rd-party dependencies. Moved spring-hibernate module's org.jsecurity.\* package to org.apache.shiro.\*.  
  
Project: shiro.txt  
Commit: f77ef5907c38280d50194b3a746a01231721cd6e  
ClassName: org.apache.shiro.samples.sprhib.web.ManageUsersController.java  
  
+ public String deleteUser(@RequestParam Long userId) {  
**+ Assert.isTrue( userId != 1, "Cannot delete admin user" );**+ userService.deleteUser( userId );  
+ return "redirect:/s/manageUsers";  
+ }  
========================================

JavaDoc updates - added convenience JavaBean properties for passthrough configuration, changed default rememberMe cookie max age due to Jetty bug  
  
Project: shiro.txt  
Commit: 0937a058b3044ed7a39762ed03029fe7cb02f7d5  
ClassName: org.jsecurity.web.WebRememberMeManager.java  
  
+ public void setCookieName(String name) {  
**+ assertCookieAttribute();**+ ((CookieAttribute) this.identityAttribute).setName(name);  
+ }  
========================================

JSEC-46 Implemented functionality for issue, also added minor updates to reference documentation  
  
Project: shiro.txt  
Commit: e8aedb64360f4449ea6dede9ebf6e3997044252f  
ClassName: org.jsecurity.mgt.SessionsSecurityManager.java  
  
+ public void setAutoCreateSessionAfterInvalidation(boolean autoCreate) {  
**+ assertSessionManager(AbstractValidatingSessionManager.class);**+ ((AbstractValidatingSessionManager) this.sessionManager).setAutoCreateAfterInvalidation(autoCreate);  
+ }  
+ public boolean isAutoCreateSessionAfterInvalidation() {  
**+ assertSessionManager(AbstractValidatingSessionManager.class);**========================================

Changes to fix problems introduced by previous refactorings.  
  
Project: shiro.txt  
Commit: 2732bd6f8e6de2ecf0d6c2fda4100eb6a91e3e79  
ClassName: org.jsecurity.ri.authc.AbstractAuthenticator.java  
  
**- private void assertCreation( AuthorizationContext authzCtx ) throws IllegalStateException {  
+ private void assertCreation( SecurityContext authzCtx ) throws IllegalStateException {** if ( authzCtx == null ) {  
- String msg = "Programming or configuration error - No AuthorizationContext was created after successful " +  
+ String msg = "Programming or configuration error - No SecurityContext was created after successful " +  
 "authentication. Verify that you have either configured the " + getClass().getName() +  
- " instance with a proper " + AuthorizationContextFactory.class.getName() + " (easier) or " +  
+ " instance with a proper " + SecurityContextFactory.class.getName() + " (easier) or " +  
 "that you have overridden the " + AbstractAuthenticator.class.getName() +  
- ".createAuthorizationContext( AuthenticationInfo info ) method.";  
+ ".createSecurityContext( AuthenticationInfo info ) method.";  
 throw new IllegalStateException( msg );  
========================================

JavaDoc updates - added convenience JavaBean properties for passthrough configuration, changed default rememberMe cookie max age due to Jetty bug  
  
Project: shiro.txt  
Commit: 0937a058b3044ed7a39762ed03029fe7cb02f7d5  
ClassName: org.jsecurity.web.WebRememberMeManager.java  
  
+ public String getCookiePath() {  
**+ assertCookieAttribute();**+ return ((CookieAttribute) this.identityAttribute).getPath();  
+ }  
========================================

Additional name change fixes related to jsecurity/ki -> shiro migration (changed case-sensitive references). Re-enabled the spring-hibernate module, cleaned up dependency management by excluding all uses of commons-logging and using jcl-over-slf4j where requried by 3rd-party dependencies. Moved spring-hibernate module's org.jsecurity.\* package to org.apache.shiro.\*.  
  
Project: shiro.txt  
Commit: f77ef5907c38280d50194b3a746a01231721cd6e  
ClassName: org.apache.shiro.samples.sprhib.web.EditUserCommand.java  
  
+ public void updateUser(User user) {  
**+ Assert.isTrue( userId.equals( user.getId() ), "User ID of command must match the user being updated." );**+ user.setUsername( getUsername() );  
+ user.setEmail( getEmail() );  
+ if( StringUtils.hasText(getPassword()) ) {  
+ user.setPassword( new Sha256Hash(getPassword()).toHex() );  
+ }  
+ }  
========================================

JavaDoc updates - added convenience JavaBean properties for passthrough configuration, changed default rememberMe cookie max age due to Jetty bug  
  
Project: shiro.txt  
Commit: 0937a058b3044ed7a39762ed03029fe7cb02f7d5  
ClassName: org.jsecurity.web.WebRememberMeManager.java  
  
+ public boolean isCookieSecure() {  
**+ assertCookieAttribute();**+ return ((CookieAttribute) this.identityAttribute).isSecure();  
+ }  
========================================

Additional name change fixes related to jsecurity/ki -> shiro migration (changed case-sensitive references). Re-enabled the spring-hibernate module, cleaned up dependency management by excluding all uses of commons-logging and using jcl-over-slf4j where requried by 3rd-party dependencies. Moved spring-hibernate module's org.jsecurity.\* package to org.apache.shiro.\*.  
  
Project: shiro.txt  
Commit: f77ef5907c38280d50194b3a746a01231721cd6e  
ClassName: org.apache.shiro.samples.sprhib.dao.HibernateUserDAO.java  
  
+ public User findUser(String username) {  
**+ Assert.hasText(username);**+ String query = "from User u where u.username = :username";  
+ return (User) getSession().createQuery(query).setString("username", username).uniqueResult();  
+ }  
========================================

SHIRO-235: implemented fix  
  
Project: shiro.txt  
Commit: 916d51ac683dc9f9f93cc50a0d9abbcc56fb9689  
ClassName: org.apache.shiro.subject.support.DelegatingSubject.java  
  
 public void checkRoles(String... roleIdentifiers) throws AuthorizationException {  
**+ assertAuthzCheckPossible();**+ securityManager.checkRoles(getPrincipals(), roleIdentifiers);  
 }  
========================================

Refactoring of the test infrastructure in order to fix XML prefix, test Locale and test TimeZone more easily.  
For this purpose, we modified MarshallerPool in order to allow the constructor to parse all properties given  
in argument, build a template for them and use the template for all (un)marshaller creations. This is likely  
to be useful not only for tests, but also in production environment.  
  
Project: sis.txt  
Commit: 06e3ba0bf78b53e095908950067b604cfd66b41d  
ClassName: org.apache.sis.xml.PooledTemplate.java  
  
+ protected void reset(final Object key, final Object value) {  
**+ throw new AssertionError();**+ }  
========================================

Refactoring of the test infrastructure in order to fix XML prefix, test Locale and test TimeZone more easily.  
For this purpose, we modified MarshallerPool in order to allow the constructor to parse all properties given  
in argument, build a template for them and use the template for all (un)marshaller creations. This is likely  
to be useful not only for tests, but also in production environment.  
  
Project: sis.txt  
Commit: 06e3ba0bf78b53e095908950067b604cfd66b41d  
ClassName: org.apache.sis.xml.PooledTemplate.java  
  
+ public Schema getSchema() {  
**+ throw new AssertionError();**+ }  
========================================

Revisit the hashCode(...) method, since we had an inconsistency with equals(...) in the way to take in account which interface is implemented.  
While fixing the inconsistency, we opportunistically lined-up the API with the equals(â€¦, ComparisonMode) one,  
in the hope to be more future-proof regarding possible SIS API evolution.  
  
Project: sis.txt  
Commit: 1b6be8c6a63df1582edaa69695d0c55d24551a6a  
ClassName: org.apache.sis.referencing.AbstractIdentifiedObject.java  
  
+ public final boolean equals(final Object object) {  
+ final boolean eq = equals(object, ComparisonMode.STRICT);  
**+ assert !eq || hashCode() == object.hashCode() : this;**+ return eq;  
+ }  
========================================

Refactoring of the test infrastructure in order to fix XML prefix, test Locale and test TimeZone more easily.  
For this purpose, we modified MarshallerPool in order to allow the constructor to parse all properties given  
in argument, build a template for them and use the template for all (un)marshaller creations. This is likely  
to be useful not only for tests, but also in production environment.  
  
Project: sis.txt  
Commit: 06e3ba0bf78b53e095908950067b604cfd66b41d  
ClassName: org.apache.sis.xml.PooledTemplate.java  
  
+ public ValidationEventHandler getEventHandler() {  
**+ throw new AssertionError();**+ }  
========================================

Support for NonSquareMatrix.solve(Matrix), and preserve DoubleDouble.error terms in NonSquareMatrix.inverse().  
Added documentation and a few more tests.  
  
Project: sis.txt  
Commit: 0b9e65ccb72fd1ceaa6624e7b885a583663a5171  
ClassName: org.apache.sis.referencing.operation.matrix.NonSquareMatrix.java  
  
+ private void copyRowFrom(final GeneralMatrix source, int srcIndex, int dstIndex) {  
+ final int numCol = this.numCol;  
**+ assert numCol == source.numCol;**+ srcIndex \*= numCol;  
+ dstIndex \*= numCol;  
+ System.arraycopy(source.elements, srcIndex, elements, dstIndex, numCol); // Copy main values  
+ System.arraycopy(source.elements, srcIndex + numCol \* source.numRow, // Copy error terms  
+ elements, dstIndex + numCol \* numRow, numCol);  
+ }  
========================================

Revisit the hashCode(...) method, since we had an inconsistency with equals(...) in the way to take in account which interface is implemented.  
While fixing the inconsistency, we opportunistically lined-up the API with the equals(â€¦, ComparisonMode) one,  
in the hope to be more future-proof regarding possible SIS API evolution.  
  
Project: sis.txt  
Commit: 1b6be8c6a63df1582edaa69695d0c55d24551a6a  
ClassName: org.apache.sis.referencing.AbstractIdentifiedObject.java  
  
- protected int computeHashCode() {  
- int code = (int) serialVersionUID;  
- for (final Class<?> type : Classes.getLeafInterfaces(getClass(), IdentifiedObject.class)) {  
- code += type.hashCode();  
+ @Override  
+ public final int hashCode() { // No need to synchronize; ok if invoked twice.  
+ int hash = hashCode;  
+ if (hash == 0) {  
+ hash = hashCode(ComparisonMode.STRICT);  
+ if (hash == 0) {  
+ hash = -1;  
 }  
- return code;  
+ hashCode = hash;  
+ }  
**+ assert hash == -1 || hash == hashCode(ComparisonMode.STRICT) : this;**+ return hash;  
 }  
========================================

Refactoring of the test infrastructure in order to fix XML prefix, test Locale and test TimeZone more easily.  
For this purpose, we modified MarshallerPool in order to allow the constructor to parse all properties given  
in argument, build a template for them and use the template for all (un)marshaller creations. This is likely  
to be useful not only for tests, but also in production environment.  
  
Project: sis.txt  
Commit: 06e3ba0bf78b53e095908950067b604cfd66b41d  
ClassName: org.apache.sis.xml.PooledTemplate.java  
  
+ public <A extends XmlAdapter> A getAdapter(final Class<A> type) {  
**+ throw new AssertionError();**+ }  
========================================

Support for NonSquareMatrix.solve(Matrix), and preserve DoubleDouble.error terms in NonSquareMatrix.inverse().  
Added documentation and a few more tests.  
  
Project: sis.txt  
Commit: 0b9e65ccb72fd1ceaa6624e7b885a583663a5171  
ClassName: org.apache.sis.referencing.operation.matrix.NonSquareMatrix.java  
  
+ private void copyColumnTo(int srcIndex, final GeneralMatrix target, int dstIndex) {  
**+ assert target.numRow == numRow;**+ while (srcIndex < elements.length) {  
+ target.elements[dstIndex] = elements[srcIndex];  
+ srcIndex += this.numCol;  
+ dstIndex += target.numCol;  
+ }  
========================================

Removed Exceptions.createParseException(...) from public API, because too convolved.  
This forced us to move LocalizedParseException to the internal package. This is a consequence  
of the need to use a different error message for the upcomming TreeTableFormat class.  
  
Project: sis.txt  
Commit: 106c25eeb1d9d81f4ebeb2e9941977c17a2b2a20  
ClassName: org.apache.sis.internal.util.LocalizedParseException.java  
  
+ private static String message(final Locale locale, Object[] arguments) {  
+ int key = specialCase(arguments[0]);  
+ if (key >= 0) {  
+ arguments = Arrays.remove(arguments, 0, 1);  
+ } else {  
+ switch (arguments.length) {  
+ case 1: key = Errors.Keys.UnexpectedEndOfString\_1; break;  
+ case 2: key = Errors.Keys.UnparsableStringForClass\_2; break;  
+ case 3: key = Errors.Keys.UnparsableStringForClass\_3; break;  
**+ default: throw new AssertionError();**+ }  
+ }  
+ return Errors.getResources(locale).getString(key, arguments);  
+ }  
========================================

SLING-2827 : several improvements/fixes:  
\* TOPOLOGY\_CHANGED is now sent with a delay after TOPOLOGY\_CHANGING - this is to avoid 'event floods' during topology changes  
\* after a TOPOLOGY\_CHANGED event is sent, a heartbeat is triggered. This speeds up change propagation. Yet it is by no means  
the case that changes now propagate as fast as possible - it is still possible for them to propage only after the  
heartbeat-interval (for downstream connector-clients)  
\* you can now configure a number of topology connectors - this replaces the 'manual' way of connecting in the webconsole  
since that is more confusing than helpful: connectors are now only created via config  
\* announcements sent by topology connectors now carry a version for paranoia compatibility checks  
\* connector servlet now accepts duplicate/loops but marks them as such - this information is also visible in the webconsole  
\* connector requests are now PUT and DELETE (rather then POST), and are done using a url of the form 'connector.\_sling-Id\_.json'  
\* logger.debug wrapped in isDebugEnabled  
\* a fix with properties changing (cannot do ModifiableHashMap.iterator.remove)  
\* a fix with proper closing of a resourceResolver to avoid unclosed sessions  
  
Project: sling.txt  
Commit: 5ef77300fa5648b47f10c68fb5f173db2c24a744  
ClassName: org.apache.sling.discovery.impl.setup.Instance.java  
  
+ private static Scheduler getSingletonScheduler() throws Exception {  
+ if (singletonScheduler!=null) {  
+ return singletonScheduler;  
+ }  
+ final Scheduler newscheduler = new QuartzScheduler();  
+ final ThreadPoolManager tpm = new DefaultThreadPoolManager(null, null);  
+ try {  
+ PrivateAccessor.invoke(newscheduler, "bindThreadPoolManager",  
+ new Class[] { ThreadPoolManager.class },  
+ new Object[] { tpm });  
+ } catch (Throwable e1) {  
**+ org.junit.Assert.fail(e1.toString());**+ }  
+ OSGiMock.activate(newscheduler);  
+ singletonScheduler = newscheduler;  
+ return singletonScheduler;  
+ }  
========================================

SLING-2822 - Rule execution uses an slf4j Logger to store output and detect issues  
  
Project: sling.txt  
Commit: 1135eab62b19c1fb733b6029ccb11a063635d82e  
ClassName: org.apache.sling.hc.it.core.U.java  
  
**- static void assertResult(EvaluationResult rr, EvaluationResult.Status status, String ruleString) {  
- assertEquals("Expecting " + rr.getRule() + " result to match", status, rr.getStatus());  
+ static void assertResult(EvaluationResult rr, boolean expectOk, String ruleString) {  
+ assertEquals("Expecting " + rr.getRule() + " result to match", expectOk, !rr.anythingToReport());  
 assertEquals("Expecting " + rr.getRule() + " string to match", ruleString, rr.getRule().toString());**========================================

STANBOL-116 STANBOL-39 :  
- org.apache.stanbol.ontologymanager.store is no longer a dependency  
- Replaced OntologyStore interface with its Clerezza implementation  
- OntologyStore class is no longer a service component  
- Duplicated OntologyStore as internal classes of ontologymanager/ontonet and reengineer/base (further decoupling will be required to solve the issue)  
- Refactored SemionManager and ONManager constructors to accept TcManager and WeightedTcProvider in their signatures instead of OntologyStore  
- Refactored jersey module accordingly  
- Unit tests now instantiate TcManager (with a JenaSparqlEngine binding) and SimpleTcProviders where needed, null values otherwise.  
  
Project: stanbol.txt  
Commit: fd9f3fc9be6b0e7a390f826e6bd6d3d112dfa027  
ClassName: eu.iksproject.kres.jersey.resource.GraphsResource.java  
  
@@ -95,3 +87,4 @@ public Response graphs(@Context HttpHeaders headers, @Context ServletContext ser  
 OWLNamedIndividual graph = factory.getOWLNamedIndividual(iri);  
**- OWLObjectPropertyAssertionAxiom axiom = factory.getOWLObjectPropertyAssertionAxiom(p, storage, graph);  
+ OWLObjectPropertyAssertionAxiom axiom = factory.getOWLObjectPropertyAssertionAxiom(p,**+ storage, graph);  
 manager.applyChange(new AddAxiom(ontology, axiom));  
========================================

fixing component version issues and adding currently refactored components to the parent pom  
  
Project: stratos.txt  
Commit: ac065d7307ff915f08d199dacbac0f0faa34ad17  
ClassName: org.apache.stratos.adc.topology.mgt.util.ConfigHolder.java  
  
+ public AxisConfiguration getAxisConfiguration() throws TopologyMgtException {  
**+ assertNull("AxisConfiguration", axisConfiguration);**+ return axisConfiguration;  
+ }  
========================================

Fixing nested group parsing issue  
  
Project: stratos.txt  
Commit: d2ebdabbc478dc676741a1b27f6e9eecf569732f  
ClassName: org.apache.stratos.autoscaler.applications.parser.DefaultApplicationParser.java  
  
@@ -646,5 +648,5 @@ private String getKillbehaviour (String serviceGroupName) throws ApplicationDefi  
**- assert serviceGroup != null;**- if (serviceGroup.getDependencies() != null) {  
- return serviceGroup.getDependencies().getTerminationBehaviour();  
**+ assert nestedServiceGroup != null;**+ if (nestedServiceGroup.getDependencies() != null) {  
+ return nestedServiceGroup.getDependencies().getTerminationBehaviour();  
 }  
========================================

fixing component version issues and adding currently refactored components to the parent pom  
  
Project: stratos.txt  
Commit: ac065d7307ff915f08d199dacbac0f0faa34ad17  
ClassName: org.apache.stratos.lb.endpoint.util.ConfigHolder.java  
  
+ public UserRegistry getConfigRegistry() throws TenantAwareLoadBalanceEndpointException {  
**+ assertNull("Registry", configRegistry);**+ return configRegistry;  
+ }  
========================================

fixing component version issues and adding currently refactored components to the parent pom  
  
Project: stratos.txt  
Commit: ac065d7307ff915f08d199dacbac0f0faa34ad17  
ClassName: org.apache.stratos.adc.topology.mgt.util.ConfigHolder.java  
  
**+ private void assertNull(String name, Object object) throws TopologyMgtException {**+ if (object == null) {  
+ String message = name + " reference in the proxy admin config holder is null";  
+ log.error(message);  
+ throw new TopologyMgtException(message);  
+ }  
+ }  
========================================

fixing component version issues and adding currently refactored components to the parent pom  
  
Project: stratos.txt  
Commit: ac065d7307ff915f08d199dacbac0f0faa34ad17  
ClassName: org.apache.stratos.lb.endpoint.util.ConfigHolder.java  
  
+ public AxisConfiguration getAxisConfiguration() throws TenantAwareLoadBalanceEndpointException {  
**+ assertNull("AxisConfiguration", axisConfiguration);**+ return axisConfiguration;  
+ }  
========================================

fixing component version issues and adding currently refactored components to the parent pom  
  
Project: stratos.txt  
Commit: ac065d7307ff915f08d199dacbac0f0faa34ad17  
ClassName: org.apache.stratos.adc.topology.mgt.util.ConfigHolder.java  
  
+ public UserRegistry getConfigRegistry() throws TopologyMgtException {  
**+ assertNull("Registry", configRegistry);**+ return configRegistry;  
+ }  
========================================

fixing component version issues and adding currently refactored components to the parent pom  
  
Project: stratos.txt  
Commit: ac065d7307ff915f08d199dacbac0f0faa34ad17  
ClassName: org.apache.stratos.lb.endpoint.util.ConfigHolder.java  
  
**+ private void assertNull(String name, Object object) throws TenantAwareLoadBalanceEndpointException {**+ if (object == null) {  
+ String message = name + " reference in the proxy admin config holder is null";  
+ log.error(message);  
+ throw new TenantAwareLoadBalanceEndpointException(message);  
+ }  
+ }  
========================================

fixing component version issues and adding currently refactored components to the parent pom  
  
Project: stratos.txt  
Commit: ac065d7307ff915f08d199dacbac0f0faa34ad17  
ClassName: org.apache.stratos.tenant.mgt.internal.util.TenantMgtRampartUtil.java  
  
+ public static Policy getDefaultRampartConfig() {  
+ ServerConfiguration serverConfig = ServerConfiguration.getInstance();  
+ String keyStore = serverConfig.getFirstProperty("Security.KeyStore.Location");  
+ String keyStoreType = serverConfig.getFirstProperty("Security.KeyStore.Type");  
+ String keyStorePassword = serverConfig.getFirstProperty("Security.KeyStore.Password");  
+ String privateKeyAlias = serverConfig.getFirstProperty("Security.KeyStore.KeyAlias");  
+ String privateKeyPassword = serverConfig.getFirstProperty("Security.KeyStore.KeyPassword");  
+ RampartConfig rampartConfig = new RampartConfig();  
+ rampartConfig.setUser(privateKeyAlias);  
+ rampartConfig.setPwCbClass("org.wso2.carbon.tenant.mgt.services.InMemoryPasswordcallbackHandler");  
+ InMemoryPasswordcallbackHandler.addUser(privateKeyAlias, privateKeyPassword);  
+ CryptoConfig sigCrypto = new CryptoConfig();  
+ Properties props = new Properties();  
+ sigCrypto.setProvider("org.apache.ws.security.components.crypto.Merlin");  
+ props.setProperty("org.apache.ws.security.crypto.merlin.keystore.type", keyStoreType);  
+ props.setProperty("org.apache.ws.security.crypto.merlin.file", keyStore);  
+ props.setProperty("org.apache.ws.security.crypto.merlin.keystore.password", keyStorePassword);  
+ sigCrypto.setProp(props);  
+ rampartConfig.setSigCryptoConfig(sigCrypto);  
+ Policy policy = new Policy();  
**+ policy.addAssertion(rampartConfig);**+ return policy;  
+ }  
========================================

fixing component version issues and adding currently refactored components to the parent pom  
  
Project: stratos.txt  
Commit: ac065d7307ff915f08d199dacbac0f0faa34ad17  
ClassName: org.apache.stratos.adc.topology.mgt.util.ConfigHolder.java  
  
+ public SynapseConfiguration getSynapseConfiguration() throws TopologyMgtException{  
**+ assertNull("SynapseConfiguration", synapseConfiguration);**+ return synapseConfiguration;  
+ }  
========================================

fixing component version issues and adding currently refactored components to the parent pom  
  
Project: stratos.txt  
Commit: ac065d7307ff915f08d199dacbac0f0faa34ad17  
ClassName: org.apache.stratos.lb.endpoint.util.ConfigHolder.java  
  
+ public SynapseConfiguration getSynapseConfiguration() throws TenantAwareLoadBalanceEndpointException{  
**+ assertNull("SynapseConfiguration", synapseConfiguration);**+ return synapseConfiguration;  
+ }  
========================================

TAP5-308: Injecting dependencies into contributeTypeCoercer method causes circular dependency error  
  
Project: tapestry5.txt  
Commit: e2c8b7b180f0418d2806de809844e1ea5f8733d4  
ClassName: org.apache.tapestry5.ioc.CyclicMOPModule.java  
  
+ public static void contributeTypeCoercer(Configuration<CoercionTuple> configuration,  
+ @Symbol("symbol-value")int value)  
+ {  
**+ assert value == 99;**+ }  
========================================

(TIKA-672) Proper error handling in the CHM parser  
  
Project: tika.txt  
Commit: 57546a3c99d2becc7e184bfeeb086b6209abcf4d  
**ClassName: org.apache.tika.parser.chm.assertion.ChmAssert.java  
- public static void assertPositiveInt(int param) {  
+ public static void assertPositiveInt(int param) throws ChmParsingException {** if (param <= 0)  
========================================

(TIKA-672) Proper error handling in the CHM parser  
  
Project: tika.txt  
Commit: 57546a3c99d2becc7e184bfeeb086b6209abcf4d  
**ClassName: org.apache.tika.parser.chm.assertion.ChmAssert.java  
- public static final void assertByteArrayNotNull(byte[] data) {  
+ public static final void assertByteArrayNotNull(byte[] data) throws ChmParsingException {** if (data == null)  
========================================

(TIKA-672) Proper error handling in the CHM parser  
  
Project: tika.txt  
Commit: 57546a3c99d2becc7e184bfeeb086b6209abcf4d  
**ClassName: org.apache.tika.parser.chm.assertion.ChmAssert.java  
- public static final void assertInputStreamNotNull(InputStream is) {  
+ public static final void assertInputStreamNotNull(InputStream is) throws IOException {** if (is == null)  
- throw new ChmParsingException("input sream is null");  
+ throw new IOException("input sream is null");  
 }  
========================================

(TIKA-672) Proper error handling in the CHM parser  
  
Project: tika.txt  
Commit: 57546a3c99d2becc7e184bfeeb086b6209abcf4d  
ClassName: org.apache.tika.parser.chm.lzx.ChmLzxBlock.java  
  
**- private void assertShortArrayNotNull(short[] array) {  
+ private void assertShortArrayNotNull(short[] array) throws TikaException {** if (array == null)  
========================================

(TIKA-672) Proper error handling in the CHM parser  
  
Project: tika.txt  
Commit: 57546a3c99d2becc7e184bfeeb086b6209abcf4d  
ClassName: org.apache.tika.parser.chm.core.ChmCommons.java  
  
@@ -90,4 +91,5 @@ public static int getWindowSize(int window) {  
 ChmLzxcResetTable resetTable, int blockNumber, int lzxcBlockOffset,  
- int lzxcBlockLength) {  
**- ChmAssert.assertChmBlockSegment(data, resetTable, blockNumber, lzxcBlockOffset, lzxcBlockLength);**+ int lzxcBlockLength) throws TikaException {  
**+ ChmAssert.assertChmBlockSegment(data, resetTable, blockNumber,**+ lzxcBlockOffset, lzxcBlockLength);  
 int blockLength = -1;  
========================================

(TIKA-672) Proper error handling in the CHM parser  
  
Project: tika.txt  
Commit: 57546a3c99d2becc7e184bfeeb086b6209abcf4d  
**ClassName: org.apache.tika.parser.chm.assertion.ChmAssert.java  
- public static void assertCopyingDataIndex(int index, int dataLength) {  
+ public static void assertCopyingDataIndex(int index, int dataLength) throws ChmParsingException {** if (index >= dataLength)  
========================================

(TIKA-672) Proper error handling in the CHM parser  
  
Project: tika.txt  
Commit: 57546a3c99d2becc7e184bfeeb086b6209abcf4d  
**ClassName: org.apache.tika.parser.chm.assertion.ChmAssert.java  
- public static final void assertChmAccessorNotNull(ChmAccessor<?> chmAccessor) {  
+ public static final void assertChmAccessorNotNull(ChmAccessor<?> chmAccessor) throws ChmParsingException {** if (chmAccessor == null)  
========================================

(TIKA-672) Proper error handling in the CHM parser  
  
Project: tika.txt  
Commit: 57546a3c99d2becc7e184bfeeb086b6209abcf4d  
ClassName: org.apache.tika.parser.chm.lzx.ChmLzxBlock.java  
  
**- private void assertInRange(short[] array, int index) {  
+ private void assertInRange(short[] array, int index) throws ChmParsingException {** if (index >= array.length)  
========================================

(TIKA-672) Proper error handling in the CHM parser  
  
Project: tika.txt  
Commit: 57546a3c99d2becc7e184bfeeb086b6209abcf4d  
ClassName: org.apache.tika.parser.chm.core.ChmCommons.java  
  
**- public static void assertByteArrayNotNull(byte[] data) {  
+ public static void assertByteArrayNotNull(byte[] data) throws TikaException {** if (data == null)  
- throw new ChmParsingException("byte[] is null");  
+ throw new TikaException("byte[] is null");  
 }  
========================================

(TIKA-672) Proper error handling in the CHM parser  
  
Project: tika.txt  
Commit: 57546a3c99d2becc7e184bfeeb086b6209abcf4d  
ClassName: org.apache.tika.parser.chm.lzx.ChmLzxBlock.java  
  
**- private void assertStateNotNull() {  
+ private void assertStateNotNull() throws TikaException {** if (getState() == null)  
========================================

Tests to show unenhanced class bugs with OpenJPA  
  
Project: tomee.txt  
Commit: 2be56e7670b5e143dd1f7049d74d070164d1130c  
ClassName: org.apache.openejb.core.cmp.jpa.UnenhancedUnits.java  
  
+ public void complexId() throws Exception {  
+ beginTx();  
+ ComplexStandalone complex = new ComplexStandalone("first", "second");  
**+ assertFalse(entityManager.contains(complex));**+ entityManager.persist(complex);  
**+ assertTrue(entityManager.contains(complex));**+ commitTx();  
+ }  
========================================

Added PrintWriter for Derby logs. Hack for setting derby.system.home. Fixed derby network startup issue  
  
Project: tomee.txt  
Commit: 13f98075732c3855f56a3afb047c6ed345c535f6  
ClassName: org.apache.openejb.alt.config.RemoteServer.java  
  
+ public static void main(String[] args) {  
**+ assert args.length > 0 : "no arguments supplied: valid arguments are 'start' or 'stop'";**+ if (args[0].equalsIgnoreCase("start")){  
+ new RemoteServer().start();  
+ } else if (args[0].equalsIgnoreCase("stop")) {  
+ RemoteServer remoteServer = new RemoteServer();  
+ remoteServer.serverHasAlreadyBeenStarted = false;  
+ remoteServer.stop();  
+ } else {  
+ throw new RuntimeException("valid arguments are 'start' or 'stop'");  
+ }  
========================================

Tests to show unenhanced class bugs with OpenJPA  
  
Project: tomee.txt  
Commit: 2be56e7670b5e143dd1f7049d74d070164d1130c  
ClassName: org.apache.openejb.core.cmp.jpa.UnenhancedUnits.java  
  
+ public void collection() throws Exception {  
+ beginTx();  
+ OneStandalone one = new OneStandalone(1000);  
+ ManyStandalone manyA = new ManyStandalone(1);  
+ one.getMany().add(manyA);  
+ manyA.setOne(one);  
+ ManyStandalone manyB = new ManyStandalone(2);  
+ one.getMany().add(manyB);  
+ manyB.setOne(one);  
+ ManyStandalone manyC = new ManyStandalone(3);  
+ one.getMany().add(manyC);  
+ manyC.setOne(one);  
**+ assertFalse(entityManager.contains(one));  
+ assertFalse(entityManager.contains(manyA));  
+ assertFalse(entityManager.contains(manyB));  
+ assertFalse(entityManager.contains(manyC));**+ entityManager.persist(one);  
+ entityManager.persist(manyA);  
+ entityManager.persist(manyB);  
+ entityManager.persist(manyC);  
+ entityManager.flush();  
**+ assertTrue(entityManager.contains(one));  
+ assertTrue(entityManager.contains(manyA));  
+ assertTrue(entityManager.contains(manyB));  
+ assertTrue(entityManager.contains(manyC));**+ commitTx();  
+ one = null;  
+ manyA = null;  
+ manyB = null;  
+ manyC = null;  
+ beginTx();  
+ one = entityManager.find(OneStandalone.class, 1000);  
**+ assertNotNull("one is null", one);  
+ assertNotNull("one.getMany() is null", one.getMany());**+ Collection<ManyStandalone> many = one.getMany();  
**+ assertEquals(3, many.size());**+ manyA = entityManager.find(ManyStandalone.class, 1);  
**+ assertNotNull("manyA is null", manyA);**+ manyB = entityManager.find(ManyStandalone.class, 2);  
**+ assertNotNull("manyB is null", manyA);**+ manyC = entityManager.find(ManyStandalone.class, 3);  
**+ assertNotNull("manyc is null", manyA);  
+ assertNotNull("manyA.getOne() is null", manyA.getOne());  
+ assertEquals(one, manyA.getOne());  
+ assertNotNull("manyB.getOne() is null", manyB.getOne());  
+ assertEquals(one, manyB.getOne());  
+ assertNotNull("manyC.getOne() is null", manyC.getOne());  
+ assertEquals(one, manyC.getOne());  
+ assertTrue(many.contains(manyA));  
+ assertTrue(many.contains(manyB));  
+ assertTrue(many.contains(manyC));**+ commitTx();  
+ }  
========================================

Fixed issue with createFoo style interfaces on local homes  
  
Project: tomee.txt  
Commit: c19ce00fab5e40dd86c9f4ce3a5c9336678324b8  
ClassName: org.apache.openejb.core.ivm.EjbHomeProxyHandler.java  
  
@@ -217,2 +216,3 @@ protected Object create(Method method, Object[] args, Object proxy) throws Throw  
 ProxyInfo proxyInfo = (ProxyInfo) container.invoke(deploymentID, method, args, null, getThreadSpecificSecurityIdentity());  
**+ assert proxyInfo != null: "Container returned a null ProxyInfo: ContainerID="+container.getContainerID();** return createProxy(proxyInfo);  
========================================

Tests to show unenhanced class bugs with OpenJPA  
  
Project: tomee.txt  
Commit: 2be56e7670b5e143dd1f7049d74d070164d1130c  
ClassName: org.apache.openejb.core.cmp.jpa.UnenhancedUnits.java  
  
+ public void generatedId() throws Exception {  
+ beginTx();  
+ GeneratedStandalone generated = new GeneratedStandalone();  
**+ assertNull("generated.getId() is not null", generated.getId());  
+ assertFalse(entityManager.contains(generated));**+ entityManager.persist(generated);  
+ entityManager.flush();  
**+ assertNotNull("generated.getId() is null", generated.getId());  
+ assertTrue(entityManager.contains(generated));**+ commitTx();  
+ }  
========================================

Updated the framework to support different key types i.e. errors , warnings and failures. Now you can specify what a key represents. If you are expecting a key to throw warnings instead of failures, you can specify it like @Keys(@Key(value="foo",type=KeyType.WARNING))  
  
Project: tomee.txt  
Commit: f694bcc731beed37fcff0bab5d569dd636878f1e  
ClassName: org.apache.openejb.config.rules.InvokeMethod.java  
  
@@ -72,4 +77,12 @@ public void evaluate() throws Throwable {  
 } catch (ValidationFailedException vfe) {  
- if (!expectedKeys.isEmpty()) {  
**- assertFailures(expectedKeys, vfe);**+ if (!isEmpty(expectedKeys)) {  
+ if (!expectedKeys.get(KeyType.FAILURE).isEmpty()) {  
**+ assertFailures(expectedKeys.get(KeyType.FAILURE), vfe);**+ }  
+ if (!expectedKeys.get(KeyType.WARNING).isEmpty()) {  
**+ assertWarnings(expectedKeys.get(KeyType.WARNING), vfe);**+ }  
+ if (!expectedKeys.get(KeyType.ERROR).isEmpty()) {  
**+ assertErrors(expectedKeys.get(KeyType.ERROR), vfe);**+ }  
 } else {  
========================================

Improved ejb-link support. Stubbed out mdb and cmp2 deployment. Fixed shutdown issue. Improved check for cmp1 finder methods.  
  
Project: tomee.txt  
Commit: 357855f89331fdce35139f4afe7b2d876a99a5a0  
ClassName: org.apache.openejb.alt.config.VmDeploymentManager.java  
  
**+ private static void assertIsDirectory(File beansDir, String variableName) {**+ if (!beansDir.exists()) {  
+ throw new IllegalArgumentException(variableName + " dir does not exist: " + beansDir);  
+ }  
+ if (!beansDir.isDirectory()) {  
+ throw new IllegalArgumentException(variableName + " dir is not a directory: " + beansDir);  
+ }  
+ }  
========================================

Tests to show unenhanced class bugs with OpenJPA  
  
Project: tomee.txt  
Commit: 2be56e7670b5e143dd1f7049d74d070164d1130c  
ClassName: org.apache.openejb.core.cmp.jpa.UnenhancedUnits.java  
  
+ public void generatedIdSubclass() throws Exception {  
+ beginTx();  
+ GeneratedSuperclass generated = new GeneratedSubclass();  
**+ assertNull("generated.getId() is not null", generated.getId());  
+ assertFalse(entityManager.contains(generated));**+ entityManager.persist(generated);  
+ entityManager.flush();  
**+ assertNotNull("generated.getId() is null", generated.getId());  
+ assertTrue(entityManager.contains(generated));**+ commitTx();  
+ }  
========================================

Tests to show unenhanced class bugs with OpenJPA  
  
Project: tomee.txt  
Commit: 2be56e7670b5e143dd1f7049d74d070164d1130c  
ClassName: org.apache.openejb.core.cmp.jpa.UnenhancedUnits.java  
  
+ public void complexIdSubclass() throws Exception {  
+ beginTx();  
+ ComplexSuperclass complex = new ComplexSubclass();  
+ complex.firstId = "first";  
+ complex.secondId = "second";  
**+ assertFalse(entityManager.contains(complex));**+ entityManager.persist(complex);  
**+ assertTrue(entityManager.contains(complex));**+ commitTx();  
+ }  
========================================

OPENEJB-210 Stateful Bean: Lifecycle Callbacks: PostConstruct/PreDestroy  
OPENEJB-213 Stateful Bean: Lifecycle Callbacks: PostActivate/PrePassivate  
Also fixed a bug in StatefulContext serialization  
  
Project: tomee.txt  
Commit: f6aee60ed8ed0922b3104660162361ec4671e396  
ClassName: org.openejb.jee.SessionBean.java  
  
+ public void addPrePassivate(String method){  
**+ assert ejbClass != null: "Set the ejbClass before calling this method";**+ getPrePassivate().add(new LifecycleCallback(ejbClass, method));  
+ }  
 public List<SecurityRoleRef> getSecurityRoleRef() {  
========================================

Cleaned up the CRUD implementation extension sample. Added comments, removed dependencies on core classes. Renamed Loader to Processor. Added FIXME comments where we need to fix things.  
  
Project: tuscany-sca-2.x.txt  
Commit: 803c9effb19d1654163593cf9dd044a2e02c6ad4  
ClassName: crud.CRUDImplementationProcessor.java  
  
+ public CRUDImplementation read(XMLStreamReader reader) throws ContributionReadException {  
**+ assert IMPLEMENTATION\_CRUD.equals(reader.getName());**+ try {  
+ String directory = reader.getAttributeValue(null, "directory");  
+ CRUDImplementation implementation = new CRUDImplementation();  
+ implementation.setDirectory(directory);  
+ while (reader.hasNext()) {  
+ if (reader.next() == END\_ELEMENT && IMPLEMENTATION\_CRUD.equals(reader.getName())) {  
+ break;  
+ }  
+ }  
+ return implementation;  
+ } catch (XMLStreamException e) {  
+ throw new ContributionReadException(e);  
+ }  
+ }  
========================================

**assertion bug fix; extension api refactors**  
Project: tuscany-sca-2.x.txt  
Commit: 8dcfbbb8942d98082bfcc145f4ce804921973017  
ClassName: org.apache.tuscany.core.extension.EntryPointContextFactory.java  
  
+ public void addSourceWireFactories(String referenceName, Class referenceInterface, List<SourceWireFactory> factories, boolean multiplicity) {  
+ if (factories.size() >1){  
+ throw new UnsupportedOperationException("Multiple wires for an entry point not allowed");  
+ }else if(factories.size() <1){  
**+ throw new AssertionError("Empty wire factory list");**+ }  
+ this.sourceWireFactory = factories.get(0);  
+ }  
========================================

Fix an issue in the connector to deal with CompositeReference (w/o binding)  
Fix the thread local issue in WorkContextImpl which shares message/correlation id across threads  
  
Project: tuscany-sca-2.x.txt  
Commit: 2cb7e7b95717487475985abbe8e1f11aa49f3f33  
ClassName: org.apache.tuscany.core.builder.ConnectorImpl.java  
  
@@ -140,2 +134,13 @@ public void connect(SCAObject source) {  
 connect(inboundWire, outboundWire, true);  
+ if (reference instanceof CompositeReference) {  
+ parent = parent.getParent();  
**+ assert parent != null : "Parent of parent was null";**+ SCAObject target = parent.getChild(outboundWire.getTargetName().getPartName());  
+ connect((Component)parent, outboundWire, target);  
+ }   
 } else if (source instanceof Service) {  
========================================

TUSCANY-611, Venkats latest fixes to the RMI binding  
  
Project: tuscany-sca-2.x.txt  
Commit: 222d29dac0ea08b39a4fd8d2873784ec00b455e3  
ClassName: org.apache.tuscany.binding.rmi.RMIInvoker.java  
  
@@ -56,4 +82,13 @@ public Object invokeTarget(Object payload) throws InvocationTargetException {  
 try {  
+ if (proxy == null) {  
+ proxy = rmiHost.findService(host,  
+ port,  
+ svcName);  
+ }  
 return remoteMethod.invoke(proxy,  
 (Object[]) payload);  
+ } catch (RemoteServiceException e) {  
**+ throw new AssertionError(e);** } catch (IllegalAccessException e) {  
========================================

Port the core to use the new set of model SPIs. There are a few test case failures to be fixed.  
  
Project: tuscany-sca-2.x.txt  
Commit: caa6aee4f9eb6b1eaf315c7d7f469866aaf2f327  
ClassName: org.apache.tuscany.core.wire.IncompatibleContractExceptionFormatter.java  
  
+ public PrintWriter write(PrintWriter writer, Throwable exception) {  
**+ assert exception instanceof IncompatibleServiceContractException;**+ IncompatibleServiceContractException e = (IncompatibleServiceContractException) exception;  
+ e.appendBaseMessage(writer);  
+ Contract source = e.getSource();  
+ String sourceContractName = null;  
+ if (source != null) {  
+ sourceContractName = source.getInterface().toString();  
+ }  
+ Operation sourceOperation = e.getSourceOperation();  
+ String sourceOpName = null;  
+ if (sourceOperation != null) {  
+ sourceOpName = sourceOperation.getName();  
+ }  
+ if (sourceOpName == null) {  
+ writer.write("\nSource Contract: " + sourceContractName);  
+ } else {  
+ writer.write("\nSource Contract: " + sourceContractName + "/" + sourceOpName);  
+ }  
+ Contract target = e.getTarget();  
+ String targetContractName = null;  
+ if (target != null) {  
+ targetContractName = target.getInterface().toString();  
+ }  
+ Operation targetOperation = e.getTargetOperation();  
+ String targetOpName = null;  
+ if (targetOperation != null) {  
+ targetOpName = targetOperation.getName();  
+ }  
+ if (targetOpName == null) {  
+ writer.write("\nTarget Contract: " + targetContractName + "\n");  
+ } else {  
+ writer.write("\nTarget Contract: " + targetContractName + "/" + targetOpName + "\n");  
+ }  
+ return writer;  
+ }  
========================================

add framework for a StAX-based loading mechanism  
still has issues with properties and references due to FIXME in ComponentImpl  
  
Project: tuscany-sca-2.x.txt  
Commit: 08d6dbf9d3c50f59652222039c98784fa8e7d532  
ClassName: org.apache.tuscany.core.loader.assembly.ServiceLoader.java  
  
+ public Service load(XMLStreamReader reader, ResourceLoader resourceLoader) throws XMLStreamException, ConfigurationLoadException {  
**+ assert SERVICE.equals(reader.getName());**+ Service service = factory.createService();  
+ service.setName(reader.getAttributeValue(null, "name"));  
+ while (true) {  
+ switch (reader.next()) {  
+ case START\_ELEMENT:  
+ AssemblyModelObject o = registry.load(reader, resourceLoader);  
+ if (o instanceof ServiceContract) {  
+ service.setServiceContract((ServiceContract) o);  
+ }  
+ reader.next();  
+ break;  
+ case END\_ELEMENT:  
+ return service;  
+ }  
+ }  
+ }  
========================================

add unit test for Operation.clone(); checkstyle fixes  
  
Project: tuscany-sca-2.x.txt  
Commit: d9db640a34ab5cc2ce01da5dc31c39d3d7b5006d  
ClassName: org.apache.tuscany.spi.model.ServiceContract.java  
  
@@ -268,3 +268,4 @@ protected Object clone() throws CloneNotSupportedException {  
 if (this.metaData != null) {  
- copy.metaData = (HashMap)this.metaData.clone();  
**+ assert this.metaData instanceof HashMap;**+ copy.metaData = (HashMap) ((HashMap) this.metaData).clone();  
 }  
========================================

add framework for a StAX-based loading mechanism  
still has issues with properties and references due to FIXME in ComponentImpl  
  
Project: tuscany-sca-2.x.txt  
Commit: 08d6dbf9d3c50f59652222039c98784fa8e7d532  
ClassName: org.apache.tuscany.core.loader.assembly.ComponentLoader.java  
  
+ public Component load(XMLStreamReader reader, ResourceLoader resourceLoader) throws XMLStreamException, ConfigurationLoadException {  
**+ assert COMPONENT.equals(reader.getName());**+ Component component = factory.createSimpleComponent();  
+ component.setName(reader.getAttributeValue(null, "name"));  
+ while (true) {  
+ switch (reader.next()) {  
+ case START\_ELEMENT:  
+ QName name = reader.getName();  
+ if (PROPERTIES.equals(name)) {  
+ loadProperties(reader, component);  
+ } else if (REFERENCES.equals(name)) {  
+ loadReferences(reader, component);  
+ } else {  
+ AssemblyModelObject o = registry.load(reader, resourceLoader);  
+ if (o instanceof ComponentImplementation) {  
+ component.setComponentImplementation((ComponentImplementation) o);  
+ }  
+ }  
+ reader.next();  
+ break;  
+ case END\_ELEMENT:  
+ return component;  
+ }  
+ }  
+ }  
========================================

clean up launch bootstrap errors and look in ../boot directory for the runtime  
  
Project: tuscany-sca-2.x.txt  
Commit: c14cf3c21b38089312e2a780aa4e030dc56ec739  
ClassName: org.apache.tuscany.launcher.MainLauncherBooter.java  
  
@@ -47,23 +42,32 @@ public static void main(String[] args) throws Throwable {  
- ClassLoader currentContextClassLoader = Thread.currentThread().getContextClassLoader();  
+ MainLauncherBooter booter = new MainLauncherBooter();  
+ ClassLoader tuscanyCL = booter.getTuscanyClassLoader();  
+ Class<?> launcherClass;  
 try {  
+ String className = System.getProperty("tuscany.LauncherClass",  
+ "org.apache.tuscany.core.launcher.MainLauncher");  
+ launcherClass = tuscanyCL.loadClass(className);  
+ } catch (ClassNotFoundException e) {  
+ System.err.println("Tuscany bootstrapper not found: " + e.getMessage());  
+ System.exit(2);  
**+ throw new AssertionError();**+ }  
- ClassLoader tuscanyCL = getTuscanyClassLoader();  
- Class<?> launcherClass = tuscanyCL.loadClass("org.apache.tuscany.core.launcher.MainLauncher");  
+ Method mainMethod;  
+ try {  
+ mainMethod = launcherClass.getMethod("main", String[].class);  
+ } catch (NoSuchMethodException e) {  
**+ throw new AssertionError(e);**+ }  
- Method mainMethod = launcherClass.getMethod("main", String[].class);  
- Thread.currentThread().setContextClassLoader(tuscanyCL);  
+ try {  
 mainMethod.invoke(null, new Object[]{args});  
- } catch (ClassNotFoundException e) {  
- System.err.println(e);  
- e.printStackTrace();  
- System.err.println("Main-Class not found: " + e.getMessage());  
- System.exit(1);  
+ } catch (IllegalAccessException e) {  
**+ throw new AssertionError(e);**+ } catch (IllegalArgumentException e) {  
**+ throw new AssertionError(e);** } catch (InvocationTargetException e) {  
- e.getCause().printStackTrace(System.err);  
- System.exit(2);  
- } finally {  
- Thread.currentThread().setContextClassLoader(currentContextClassLoader);  
+ throw e.getCause();  
 }  
========================================

checkin fix for TUSCANY-40; continued entrypoint implementation, which will not work yet  
  
Project: tuscany-sca-2.x.txt  
Commit: 7b28ccc3ffcfc2b1171862b7c643c743dffe7282  
ClassName: org.apache.tuscany.core.builder.impl.EntryPointRuntimeConfiguration.java  
  
+ public void addSourceProxyFactory(String referenceName, ProxyFactory factory) {  
**+ assert (referenceName != null) : "No reference name specified";  
+ assert (factory != null) : "Proxy factory was null";**+ this.epServiceName = referenceName; // entry points are configured with only one reference  
+ this.proxyFactory = factory;  
+ }  
========================================

[UIMA-2544] Apply UIMA code style  
https://issues.apache.org/jira/browse/UIMA-2544  
- Applied Eclipse's "Format code" function  
  
Project: uima-uimafit.txt  
Commit: 8548d69b7278c074ac9166b3fde557d7c30c200b  
ClassName: org.apache.uima.fit.util.CasUtil.java  
  
@@ -546,8 +538,7 @@ public static Type getAnnotationType(CAS aCas, String aTypeName) {  
**- assert (a.getBegin() >= begin) : "Illegal begin " + a.getBegin() + " in [" + begin**- + ".." + end + "]";  
**- assert (a.getEnd() <= end) : "Illegal end " + a.getEnd() + " in [" + begin + ".."  
+ assert (a.getBegin() >= begin) : "Illegal begin " + a.getBegin() + " in [" + begin + ".."** + end + "]";  
**+ assert (a.getEnd() <= end) : "Illegal end " + a.getEnd() + " in [" + begin + ".." + end + "]";** list.add(a);  
========================================

UIMA-48: reformatted code and fixed trivial compiler warnings  
  
Project: uima-uimaj.txt  
Commit: dbaff31162e71340807de9f506ab89d5af117437  
ClassName: org.apache.uima.cas\_data.impl.CasComparer.java  
  
**- public static void assertEquals(CAS c1, CAS c2)**- {  
+public class CasComparer {  
**+ public static void assertEquals(CAS c1, CAS c2) {** FSIterator sofaIter1 = c1.getSofaIterator();  
 FSIterator sofaIter2 = c2.getSofaIterator();  
- while (sofaIter1.hasNext())  
- {  
+ while (sofaIter1.hasNext()) {  
 **Assert.assertTrue(sofaIter2.hasNext());**========================================

UIMA-48: reformatted code and fixed trivial compiler warnings  
  
Project: uima-uimaj.txt  
Commit: dbaff31162e71340807de9f506ab89d5af117437  
ClassName: org.apache.uima.cas\_data.impl.CasComparer.java  
  
**@@ -61,4 +59,3 @@ public static void assertEquals(CAS c1, CAS c2)  
- public static void assertEquals(TCAS c1, TCAS c2)**- {  
**+ public static void assertEquals(TCAS c1, TCAS c2) {** HashMap visited = new HashMap();  
========================================

UIMA-48: reformatted code and fixed trivial compiler warnings  
  
Project: uima-uimaj.txt  
Commit: dbaff31162e71340807de9f506ab89d5af117437  
ClassName: org.apache.uima.cas\_data.impl.CasComparer.java  
  
**@@ -84,11 +79,7 @@ public static void assertEquals(FeatureStructure fs1, FeatureStructure fs2)  
- public static void assertEquals(FeatureStructure fs1, FeatureStructure fs2, Map visited)**- {  
- if (fs1 == null)  
- {  
**+ public static void assertEquals(FeatureStructure fs1, FeatureStructure fs2, Map visited) {**+ if (fs1 == null) {  
 **Assert.assertNull(fs2);** return;  
- }  
- else  
- {  
+ } else {  
 **Assert.assertNotNull(fs2);**========================================

XMI CAS Serializer/Deserializer updates for merging and  
out-of-typesystem data.  
UIMA-325: http://issues.apache.org/jira/browse/UIMA-325  
UIMA-326: http://issues.apache.org/jira/browse/UIMA-326  
  
Project: uima-uimaj.txt  
Commit: fa2231b3ca3f19859b913f7c5b2b6702d6e0ba85  
ClassName: org.apache.uima.cas.impl.ListUtils.java  
  
@@ -257,4 +259,20 @@ public boolean isListType(int type) {  
 int heapVal = cas.getHeapValue(curNode + cas.getFeatureOffset(fsHeadFeat));  
+ if (heapVal == 0) {  
+ strArray[i] = "0";  
+ if (sharedData != null) {  
+ OotsElementData oed = sharedData.getOutOfTypeSystemFeatures(curNode);  
+ if (oed != null) {  
**+ assert oed.attributes.size() == 1; //only the head feature can possibly be here**+ XmlAttribute attr = (XmlAttribute)oed.attributes.get(0);  
**+ assert CAS.FEATURE\_BASE\_NAME\_HEAD.equals(attr.name);**+ strArray[i] = attr.value;  
+ }  
+ }   
+ }  
+ else {  
 if (sharedData != null) {  
- strArray[i] = sharedData.getXmiId(heapVal);  
+ strArray[i] = heapVal == 0 ? null : sharedData.getXmiId(heapVal);  
 } else {  
========================================

Added source CAS as argument to CasCopier constructor.  
UIMA-215: https://issues.apache.org/jira/browse/UIMA-215  
  
Project: uima-uimaj.txt  
Commit: c0cb21b254897eab457f827635be4e83f8b2137d  
ClassName: org.apache.uima.util.CasCopier.java  
  
 public FeatureStructure copyFs(FeatureStructure aFS) {  
**+ assert ((CASImpl)aFS.getCAS()).getBaseCAS() == ((CASImpl)mSrcCas).getBaseCAS();**========================================

CASes available to CAS Multipliers now have identical TypeSystem  
objects as CASes in the CPE CAS Pool.  
UIMA-234: https://issues.apache.org/jira/browse/UIMA-234  
  
Project: uima-uimaj.txt  
Commit: f92ea4923c1fb8b8a8b86bcac66305729ccaf7a7  
ClassName: org.apache.uima.collection.impl.SingleDocCollectionReader.java  
  
+ public void getNext(CAS aCAS) throws IOException, CollectionException {  
**+ assert(!done);**+ aCAS.setDocumentText("This is a test");  
+ done = true;  
+ }  
========================================

UIMA-48: reformatted code and fixed trivial compiler warnings  
  
Project: uima-uimaj.txt  
Commit: dbaff31162e71340807de9f506ab89d5af117437  
ClassName: org.apache.uima.cas\_data.impl.CasComparer.java  
  
**@@ -79,4 +75,3 @@ public static void assertEquals(TCAS c1, TCAS c2)  
- public static void assertEquals(FeatureStructure fs1, FeatureStructure fs2)**- {  
**+ public static void assertEquals(FeatureStructure fs1, FeatureStructure fs2) {  
 assertEquals(fs1, fs2, new HashMap());**========================================

Implemented the creation of a default bag index when an attempt  
is made to add an FS to the indexes when there are no indexes  
defined over that FS's type.  
UIMA-173:http://issues.apache.org/jira/browse/UIMA-173  
  
Project: uima-uimaj.txt  
Commit: 49259d29de3a1a71f9a0352da18711a631eece65  
ClassName: org.apache.uima.cas.impl.FSIndexRepositoryImpl.java  
  
@@ -1352,3 +1337,11 @@ public void ll\_addFS(int fsRef) {  
 if (size == 0) {  
- this.undefinedIndex.add(fsRef);  
+ Type type = this.typeSystem.getType(typeCode);  
+ String defIndexName = "\_" + type.getName() + "\_GeneratedIndex";  
+ FSIndexComparator comparator = createComparator();  
+ comparator.setType(type);  
+ createIndexNoQuestionsAsked(comparator, defIndexName, FSIndex.BAG\_INDEX);  
**+ assert this.indexArray[typeCode].size() == 1;**+ ((IndexIteratorCachePair)this.indexArray[typeCode].get(0)).index.insert(fsRef);  
 }  
========================================

Fixed AbstractRestIT so that it would no longer use aquireToken calls.  
Fixed before in TestContextSetup so that we create unique test accounts each time  
Reformatted and fixed every single test in OrganizationsIT  
Added a uuid when TestUser gets created.  
  
Project: usergrid.txt  
Commit: 0fc8818b8135630c7ec6b2725a77a96923c6c97b  
ClassName: org.apache.usergrid.rest.management.OrganizationsIT.java  
  
@@ -98,27 +111,6 @@ public void createOrgAndOwner() throws Exception {  
- UserInfo ui = setup.getMgmtSvc().getAdminUserByEmail( "test-user-1@mockserver.com" );  
 EntityManager em = setup.getEmf().getEntityManager( setup.getEmf().getManagementAppId() );  
- User user = em.get( ui.getUuid(), User.class );  
**- assertEquals( "Test User", user.getName() );**+ User user = em.get( userUuid, User.class );  
**+ assertEquals( name, user.getName() );  
 assertEquals( "Apigee", user.getProperty( "company" ) );**- OrganizationInfo orgInfo = setup.getMgmtSvc().getOrganizationByName( "test-org-1" );  
**- assertEquals( 5L, orgInfo.getProperties().get( "securityLevel" ) );**- node = mapper.readTree( resource().path( "/management/organizations/test-org-1" )  
- .queryParam( "access\_token", superAdminToken() ).accept( MediaType.APPLICATION\_JSON )  
- .type( MediaType.APPLICATION\_JSON\_TYPE ).get( String.class ));  
- logNode( node );  
**- Assert.assertEquals( 5, node.get( "organization" ).get( OrganizationsResource.ORGANIZATION\_PROPERTIES )**- .get( "securityLevel" ).asInt() );  
- node = mapper.readTree( resource().path( "/management/organizations/test-org-1" )  
- .queryParam( "access\_token", superAdminToken() ).accept( MediaType.APPLICATION\_JSON )  
- .type( MediaType.APPLICATION\_JSON\_TYPE ).get( String.class ));  
**- Assert.assertEquals( 5, node.get( "organization" ).get( OrganizationsResource.ORGANIZATION\_PROPERTIES )**- .get( "securityLevel" ).asInt() );  
- }  
========================================

USERGRID-2625:  
o Added JobListener interface for callbacks to scheduler  
o Added setters to register JobListener  
o Created TestJobListener utility class for to collect stats  
o Created test case for TestJobListener  
o Used TestJobListener to fix and make deterministic SchedularRuntimeXIT tests  
o Cleaned up a few things here and there and to comply with ASF requirements  
o Added AL 2.0 Headers and removed @author tags  
o Cleaned up imports  
o Added package-info.java files for some packages  
  
Project: usergrid.txt  
Commit: c477e8d67836b7097f08169d39197040e6ff6bac  
ClassName: org.usergrid.batch.job.SchedulerRuntime2IT.java  
  
@@ -65,5 +59,6 @@ public void schedulingWithNoJobs() throws InterruptedException {  
- waited = counterJob.waitForCount( 3000000, TimeUnit.SECONDS );  
+ waited = getJobListener().blockTilDone( 2 \* getCount(), 5000L );  
 **assertTrue( "Jobs ran", waited );  
+ assertTrue( 2 \* getCount() + " successful jobs ran",**+ ( 2 \* getCount() ) == getJobListener().getSuccessCount() );  
 }  
========================================

fix build error  
  
Project: usergrid.txt  
Commit: e30a7b32258b95b1d5d74d6c9ba448dcc80a42f1  
ClassName: org.apache.usergrid.services.notifications.apns.NotificationsServiceIT.java  
  
@@ -733,7 +722,5 @@ public void inactiveDeviceUpdate() throws Exception {  
 device1 = app.getEm().get(device1, Device.class);  
**- assertNull(device1**- .getProperty(notifier.getName() + NOTIFIER\_ID\_POSTFIX));  
**+ assertNull(device1 .getProperty(notifier.getName() + NOTIFIER\_ID\_POSTFIX));** device2 = app.getEm().get(device2, Device.class);  
**- assertNull(device2**- .getProperty(notifier.getName() + NOTIFIER\_ID\_POSTFIX));  
**+ assertNull(device2 .getProperty(notifier.getName() + NOTIFIER\_ID\_POSTFIX));** }  
========================================

Fixed read bug  
  
Project: usergrid.txt  
Commit: 1bb4d55e24872971e38e3420a2e9ce21c8ce906c  
ClassName: org.usergrid.tools.EntityReadBenchMark.java  
  
@@ -305,3 +319,3 @@ protected void doRead(String value) {  
**- Assert.isTrue(indexer.existsInIndex(appId, "test", "test", value));  
+ Assert.isTrue(indexer.existsInIndex(appId, "tests", "test", value));**========================================

Fixes rolling version bug when merge occurs.  
  
Project: usergrid.txt  
Commit: 2828e911e5888e5a2f26ff412c3182c22cbf09c2  
ClassName: org.apache.usergrid.persistence.collection.EntityCollectionManagerIT.java  
  
@@ -301,2 +312,6 @@ public void partialUpdate() {  
+ final UUID returnedVersion = newEntity.getVersion();  
**+ assertEquals("Loaded version matches updated version", updatedVersion, returnedVersion);  
 assertNotNull( "Returned has a uuid", returned.getId() );**========================================

Added in fix for USERGRID-2019. Added test to confirm issue is fixed.  
  
Project: usergrid.txt  
Commit: aa522b0e9d8186bf468588a48b69c4196cba6123  
ClassName: org.usergrid.rest.management.users.MUUserResourceIT.java  
  
+ public void passwordResetIncorrectUserName() throws Exception {  
+ String email = "test@usergrid.com";  
+ UserInfo userInfo = setup.getMgmtSvc().getAdminUserByEmail(email);  
+ String resetToken = setup.getMgmtSvc().getPasswordResetTokenForAdminUser(userInfo.getUuid(), 15000);  
**+ assertTrue(setup.getMgmtSvc().checkPasswordResetTokenForAdminUser(userInfo.getUuid(), resetToken));**+ Form formData = new Form();  
+ formData.add("token", resetToken);  
+ formData.add("password1", "sesame");  
+ formData.add("password2", "sesame");  
+ String html = resource()  
+ .path("/management/users/" +"noodle"+userInfo.getUsername() + "/resetpw")  
+ .type(MediaType.APPLICATION\_FORM\_URLENCODED\_TYPE)  
+ .post(String.class, formData);  
**+ assertTrue(html.contains("Incorrect username entered"));**+ html = resource()  
+ .path("/management/users/" +userInfo.getUsername() + "/resetpw")  
+ .type(MediaType.APPLICATION\_FORM\_URLENCODED\_TYPE)  
+ .post(String.class, formData);  
**+ assertTrue(html.contains("password set"));**+ }  
+ @Test  
 public void checkPasswordHistoryConflict() throws Exception {  
========================================

USERGRID-1868, USERGRID-1874  
o Fix minor issues with poms due to some IDEA plugins not pulling transitive  
dependencies properly  
o Added the standard unit, default, unit-concurrent, integ-concurrent Maven  
profiles to the schedular module  
o Broke apart a disabled test cases for the SchedularRuntimeTest which is a  
very long running integration test and used both Maven Surefire parallel  
tags and the ConcurrentRunner to speed it up once Todd re-enables it. The  
tests were passing for me with full isolation using just Maven Surefire  
parallelism with a CassandraResource allocated to each test case with a  
new JVM.  
  
Project: usergrid.txt  
Commit: c2507e27298a8fff91ab47efee73840ae86f3422  
ClassName: org.usergrid.batch.job.SchedulerRuntime2IT.java  
  
+ public void schedulingWithNoJobs() throws InterruptedException {  
+ int count = 200;  
+ CountdownLatchJob counterJob = cassandraResource.getBean(CountdownLatchJob.class);  
+ counterJob.setLatch(count);  
+ for (int i = 0; i < count; i++) {  
+ scheduler.createJob("countdownLatch", System.currentTimeMillis(), new JobData());  
+ }  
+ boolean waited = counterJob.waitForCount(30000, TimeUnit.SECONDS);  
**+ assertTrue("Jobs ran", waited);**+ Thread.sleep(5000);  
+ counterJob.setLatch(count);  
+ for (int i = 0; i < count; i++) {  
+ scheduler.createJob("countdownLatch", System.currentTimeMillis(), new JobData());  
+ }  
+ waited = counterJob.waitForCount(3000000, TimeUnit.SECONDS);  
**+ assertTrue("Jobs ran", waited);**+ }  
========================================

Fixed collections bug, fixed all tests to prepare for removal of ExportInfo.  
  
Project: usergrid.txt  
Commit: 9e716a6f6c28639b77b39c239ded544ff2f4873e  
ClassName: org.apache.usergrid.rest.management.ManagementResourceIT.java  
  
+ @Test  
+ public void exportGetWrongUUID() throws Exception {  
+ JsonNode node = null;  
+ Status responseStatus = Status.OK;  
+ UUID fake = UUID.fromString( "AAAAAAAA-FFFF-FFFF-FFFF-AAAAAAAAAAAA" );  
+ try {  
+ node = resource().path( "/management/orgs/test-organization/apps/test-app/export/" + fake )  
+ .queryParam( "access\_token", superAdminToken() ).accept( MediaType.APPLICATION\_JSON )  
+ .type( MediaType.APPLICATION\_JSON\_TYPE ).get( JsonNode.class );  
+ }  
+ catch ( UniformInterfaceException uie ) {  
+ responseStatus = uie.getResponse().getClientResponseStatus();  
+ }  
**+ assertEquals( Status.BAD\_REQUEST, responseStatus );**+ }  
========================================

Fixing merge issues in query validator, still fighting a compile error in the tests.  
  
Project: usergrid.txt  
Commit: 3dc1e9fe8e3cd0c7a25576cacd8be567391588f9  
ClassName: org.apache.usergrid.query.validator.users.UserQueryIT.java  
  
+ public void nameBeginswithAndSexEqualAndAgeGreaterthanequalOrSexEqual\_limitL20() {  
+ String sqlite = "SELECT \* FROM users WHERE name LIKE 'a%' and sex = 'male' and age >= 20 or sex = 'female' LIMIT 20";  
+ String api = "select \* where name = 'a\*' and sex = 'male' and age >= 20 or sex = 'female'";  
+ int limit = 20;  
+ QueryRequest request = new QueryRequest();  
+ request.setDbQuery(sqlite);  
+ request.getApiQuery().setQuery(api);  
+ request.getApiQuery().setLimit(limit);  
+ QueryResponse response = validator.execute(request);  
**+ Assert.assertTrue(response.toString(), response.result());**+ }  
========================================

Fixing merge issues in query validator, still fighting a compile error in the tests.  
  
Project: usergrid.txt  
Commit: 3dc1e9fe8e3cd0c7a25576cacd8be567391588f9  
ClassName: org.apache.usergrid.query.validator.users.UserQueryIT.java  
  
+ public void nameBeginswithAndSexEqualAndAgeGreaterthanequalOrSexEqual\_sortAddressAscNameDesc() {  
+ String sqlite = "SELECT \* FROM users WHERE name LIKE 'a%' and sex = 'male' and age >= 35 or sex = 'female' ORDER BY address asc, name desc LIMIT 4";  
+ String api = "select \* where name = 'a\*' and sex = 'male' and age >= 35 or sex = 'female' order by address asc, name desc";  
+ QueryRequest request = new QueryRequest();  
+ request.setDbQuery(sqlite);  
+ request.getApiQuery().setQuery(api);  
+ request.getApiQuery().setLimit(4);  
+ QueryResponse response = validator.execute(request);  
**+ Assert.assertTrue(response.toString(), response.result());**+ }  
========================================

resolve misc serialization issues (FindBugs)  
  
Project: velocity-tools.txt  
Commit: 3b501e94318508248207a230ac0f5e4f8038ed36  
ClassName: org.apache.velocity.tools.view.VelocityViewServlet.java  
  
@@ -140,13 +129,22 @@ protected String findInitParameter(ServletConfig config, String key)  
- protected void setVelocityView(VelocityView view)  
+ protected VelocityView getVelocityView()  
 {  
- if (view == null)  
+ if (this.view == null)  
 {  
- throw new NullPointerException("VelocityView cannot be null");  
- }  
- this.view = view;  
+ ServletConfig config = getServletConfig();  
+ String shared = findInitParameter(config, SHARED\_CONFIG\_PARAM);  
+ if (shared == null || shared.equals("false"))  
+ {  
+ this.view = ServletUtils.getVelocityView(config);  
**+ assert (view != null);** }  
========================================

Improved the error reporting in test cases based on XMLStreamReaderComparator.  
  
Project: webservices-axiom.txt  
Commit: d0c5d2b4cd470527942eba3a40c0f07e4683a146  
ClassName: org.apache.axiom.stax.XMLStreamReaderComparator.java  
  
**@@ -102,4 +115,6 @@ private Object assertSameResult(String methodName, Class[] paramTypes, Object[]** if (results != null) {  
**- assertEquals("Return value of " + methodName + " (event type " +**- StAXUtils.getEventTypeString(expected.getEventType()) + ")",  
**+ assertEquals("Return value of " + methodName + " for arguments " +**+ Arrays.asList(args) + " (event type " +  
+ StAXUtils.getEventTypeString(expected.getEventType()) +  
+ "; location " + getLocation() + ")",  
 results[0], results[1]);  
========================================

Extended the coverage of XMLStreamReaderComparator and fixed OMStAXWrapper#getNamespaceContext().  
  
Project: webservices-axiom.txt  
Commit: 88d66f685084c8475070a991b7807e87e6cba251  
ClassName: org.apache.axiom.stax.XMLStreamReaderComparator.java  
  
+ private void compareNamespaceContexts(NamespaceContext expected, NamespaceContext actual) {  
+ for (Iterator it = prefixes.iterator(); it.hasNext(); ) {  
+ String prefix = (String)it.next();  
+ if (prefix != null) {  
**+ assertEquals("Namespace URI for prefix '" + prefix + "' (" + getLocation() + ")", expected.getNamespaceURI(prefix), actual.getNamespaceURI(prefix));**+ }  
+ }  
========================================

Some refactoring and few bug fixes ..  
  
Project: webservices-neethi.txt  
Commit: a2e8a5afc11cb85322ed526d91aeade34523c470  
ClassName: org.apache.policy.util.PolicyWriter.java  
  
**@@ -95,4 +95,4 @@ private void writeAndCompositeAssertion(AndCompositeAssertion assertion,** XMLStreamWriter writer) throws XMLStreamException {  
- writer.writeStartElement(WSPConstants.WS\_POLICY\_PREFIX,   
**- WSPConstants.AND\_COMPOSITE\_ASSERTION, WSPConstants.WS\_POLICY\_NAMESPACE\_URI);**+ writer.writeStartElement(PolicyConstants.WS\_POLICY\_PREFIX,   
**+ PolicyConstants.AND\_COMPOSITE\_ASSERTION, PolicyConstants.WS\_POLICY\_NAMESPACE\_URI);**========================================

Some refactoring and few bug fixes ..  
  
Project: webservices-neethi.txt  
Commit: a2e8a5afc11cb85322ed526d91aeade34523c470  
ClassName: org.apache.policy.util.PolicyWriter.java  
  
**@@ -106,4 +106,4 @@ private void writeXorCompositeAssertion(XorCompositeAssertion assertion,** XMLStreamWriter writer) throws XMLStreamException {  
- writer.writeStartElement(WSPConstants.WS\_POLICY\_PREFIX,   
**- WSPConstants.XOR\_COMPOSITE\_ASSERTION, WSPConstants.WS\_POLICY\_NAMESPACE\_URI);**+ writer.writeStartElement(PolicyConstants.WS\_POLICY\_PREFIX,   
**+ PolicyConstants.XOR\_COMPOSITE\_ASSERTION, PolicyConstants.WS\_POLICY\_NAMESPACE\_URI);**========================================

Fix spelling mistake in class name  
Add preliminary support for WS-Policy 1.5  
**Add ignorable attribute to assertions (for 1.5)**  
Project: webservices-neethi.txt  
Commit: 4d942127e5e900e399699a17ae0184675b934087  
**ClassName: org.apache.neethi.builders.xml.XmlPrimitiveAssertion.java**+ public boolean equal(PolicyComponent policyComponent) {  
**+ if (policyComponent.getType() != Constants.TYPE\_ASSERTION) {**+ return false;  
+ }  
**+ return getName().equals(((Assertion) policyComponent).getName());**+ }  
========================================

Some refactoring and few bug fixes ..  
  
Project: webservices-neethi.txt  
Commit: a2e8a5afc11cb85322ed526d91aeade34523c470  
ClassName: org.apache.policy.util.PolicyUtil.java  
  
+ public static Policy getPolicy(List terms) {  
+ Policy policyTerm = new Policy();  
**+ XorCompositeAssertion xorTerm = new XorCompositeAssertion();  
+ AndCompositeAssertion andTerm = new AndCompositeAssertion();**+ andTerm.addTerms(terms);  
+ xorTerm.addTerm(andTerm);  
+ policyTerm.addTerm(xorTerm);  
+ return policyTerm;  
+ }  
========================================

Fixed: normalize(..) causes an infinte recurrsion which result in stack overflow error  
  
Project: webservices-neethi.txt  
Commit: 50aefeeb1250781a2a5561c9d552a098ff53879a  
**ClassName: org.apache.ws.policy.PrimitiveAssertion.java  
@@ -220,56 +222,100 @@ public Assertion normalize(PolicyRegistry reg) {**- Iterator iterator2 = xorTerm.getTerms().iterator();  
- if (!(iterator2.hasNext())) { //policy with no alternatives  
+ List ANDs = xorTerm.getTerms();  
- if (childNonPolicyTerms.isEmpty()) {  
**- PrimitiveAssertion primTerm = getSelfWithoutTerms();**- primTerm.addTerm(policyTerm);  
- return primTerm;   
+ if (ANDs.size() == 0) {  
**+ return new XorCompositeAssertion();** }  
- ArrayList allTerms = new ArrayList();  
- allTerms.addAll(childNonPolicyTerms);  
**- allTerms.addAll(((AndCompositeAssertion) iterator2.next()).getTerms());  
- PrimitiveAssertion primTerm = getSelfWithoutTerms();**- primTerm.addTerm(getSinglePolicy(allTerms));  
- return primTerm;   
+ if (ANDs.size() == 1) {  
**+ ((AndCompositeAssertion) ANDs.get(0)).addTerms(nonPolicyTerms);  
+ PrimitiveAssertion PRIM = getSelfWithoutTerms();**+ PRIM.addTerm(policyTerm);  
+ return PRIM;  
 }  
- Policy endPolicyTerm = new Policy();  
**- XorCompositeAssertion endXorTerm = new XorCompositeAssertion();**- endPolicyTerm.addTerm(endXorTerm);   
- ArrayList endAndTerms = new ArrayList();  
**- AndCompositeAssertion anEndAndTerm = new AndCompositeAssertion();  
- PrimitiveAssertion self;**+ Policy nPOLICY = new Policy();  
**+ XorCompositeAssertion nXOR = new XorCompositeAssertion();**+ nPOLICY.addTerm(nXOR);  
- ArrayList termsForAnEndAnd = new ArrayList();  
- self = getSelfWithoutTerms();  
**+ PrimitiveAssertion nPRIM;**+ Iterator iterator2 = ANDs.iterator();  
- termsForAnEndAnd.addAll(childNonPolicyTerms);  
**- termsForAnEndAnd.addAll(((AndCompositeAssertion) iterator2.next()).getTerms());**- self.addTerm(PolicyUtil.getPolicy(termsForAnEndAnd));  
- anEndAndTerm.addTerm(self);  
- endAndTerms.add(anEndAndTerm);  
+ ArrayList list;  
 while (iterator2.hasNext()) {  
**- anEndAndTerm = new AndCompositeAssertion();**- self = getSelfWithoutTerms();  
+ nPRIM = getSelfWithoutTerms();  
- termsForAnEndAnd.clear();  
- termsForAnEndAnd.addAll(childNonPolicyTerms);  
**- termsForAnEndAnd.addAll(((AndCompositeAssertion) iterator2.next()).getTerms());**- self.addTerm(PolicyUtil.getPolicy(termsForAnEndAnd));  
+ list = new ArrayList();  
**+ list.addAll(((AndCompositeAssertion) iterator2.next()).getTerms());**- anEndAndTerm.addTerm(self);  
- endAndTerms.add(anEndAndTerm);   
+ if (!nonPolicyTerms.isEmpty()) {  
+ list.addAll(nonPolicyTerms);   
+ }  
+ nPRIM.addTerm(getSinglePolicy(list));  
**+ AndCompositeAssertion AND = new AndCompositeAssertion();**+ AND.addTerm(nPRIM);  
+ nXOR.addTerm(AND);   
 }  
- endXorTerm.addTerms(endAndTerms);  
- return endPolicyTerm;  
+ nPOLICY.setNormalized(true);  
+ return nPOLICY;  
 }  
========================================

Fix spelling mistake in class name  
Add preliminary support for WS-Policy 1.5  
**Add ignorable attribute to assertions (for 1.5)**  
Project: webservices-neethi.txt  
Commit: 4d942127e5e900e399699a17ae0184675b934087  
**ClassName: org.apache.neethi.builders.xml.XMLPrimitiveAssertionBuilder.java  
@@ -51,5 +62,5 @@ public Assertion build(OMElement element, AssertionBuilderFactory factory)** Policy policy = PolicyEngine.getPolicy(el);  
**- return new NestedPrimitiveAssertion(element.getQName(), isOptional, policy);  
+ return new PolicyContainingAssertion(element.getQName(), optional, ignorable, policy);** }  
**- return new XmlPrimtiveAssertion(element);  
+ return new XmlPrimitiveAssertion(element);** }  
========================================

Some refactoring and few bug fixes ..  
  
Project: webservices-neethi.txt  
Commit: a2e8a5afc11cb85322ed526d91aeade34523c470  
ClassName: org.apache.policy.util.PolicyReader.java  
  
**@@ -73,12 +73,12 @@ private Assertion readAssertion(OMElement element) {**- if (localName.equals(WSPConstants.WS\_POLICY)) {  
+ if (localName.equals(PolicyConstants.WS\_POLICY)) {  
 return readPolicy(element);  
**- } else if (localName.equals(WSPConstants.AND\_COMPOSITE\_ASSERTION)) {  
+ } else if (localName.equals(PolicyConstants.AND\_COMPOSITE\_ASSERTION)) {** return readAndComposite(element);  
**- } else if (localName.equals(WSPConstants.XOR\_COMPOSITE\_ASSERTION)) {  
+ } else if (localName.equals(PolicyConstants.XOR\_COMPOSITE\_ASSERTION)) {** return readXorComposite(element);  
- } else if (localName.equals(WSPConstants.WS\_POLICY\_REFERENCE)) {  
+ } else if (localName.equals(PolicyConstants.WS\_POLICY\_REFERENCE)) {  
 return readPolicyReference(element);  
========================================

Fix spelling mistake in class name  
Add preliminary support for WS-Policy 1.5  
**Add ignorable attribute to assertions (for 1.5)**  
Project: webservices-neethi.txt  
Commit: 4d942127e5e900e399699a17ae0184675b934087  
**ClassName: org.apache.neethi.builders.PolicyContainingAssertion.java**+ public PolicyComponent normalize() {  
+ Policy normalisedNested   
+ = (Policy)nested.normalize(true);  
+ Policy p = new Policy();  
+ ExactlyOne ea = new ExactlyOne();  
+ p.addPolicyComponent(ea);  
+ if (isOptional()) {  
+ ea.addPolicyComponent(new All());  
+ }  
+ Iterator<List<PolicyComponent>> alternatives = normalisedNested.getAlternatives();  
+ while (alternatives.hasNext()) {  
+ All all = new All();  
+ List<PolicyComponent> alternative = alternatives.next();  
+ Policy n = new Policy();  
**+ PolicyContainingAssertion a   
+ = new PolicyContainingAssertion(getName(), false, ignorable, nested);**+ ExactlyOne nea = new ExactlyOne();  
+ n.addPolicyComponent(nea);  
+ All na = new All();  
+ nea.addPolicyComponent(na);  
+ na.addPolicyComponents(alternative);  
+ all.addPolicyComponent(a);  
+ ea.addPolicyComponent(all);   
+ }   
+ return p;   
+ }   
========================================

Fix spelling mistake in class name  
Add preliminary support for WS-Policy 1.5  
**Add ignorable attribute to assertions (for 1.5)**  
Project: webservices-neethi.txt  
Commit: 4d942127e5e900e399699a17ae0184675b934087  
**ClassName: org.apache.neethi.builders.PolicyContainingAssertion.java**+ public boolean equal(PolicyComponent policyComponent) {  
+ if (!super.equal(policyComponent)) {  
+ return false;  
+ }  
**+ PolicyContainingAssertion other = (PolicyContainingAssertion)policyComponent;**+ return getPolicy().equal(other.getPolicy());  
+ }  
========================================

Some refactoring and few bug fixes ..  
  
Project: webservices-neethi.txt  
Commit: a2e8a5afc11cb85322ed526d91aeade34523c470  
ClassName: org.apache.policy.util.PolicyUtil.java  
  
+ public static List getPrimTermsList(Policy policy) {  
+ if (!policy.isNormalized()) {  
+ policy = (Policy) policy.normalize();  
+ }  
**+ XorCompositeAssertion xorTerm   
+ = (XorCompositeAssertion) policy.getTerms().get(0);  
+ AndCompositeAssertion andTerm   
+ = (AndCompositeAssertion) xorTerm.getTerms().get(0);**+ return andTerm.getTerms();  
+ }  
========================================

- fixed test case  
- some more changes to the "framework"  
Will adjust and apply Pedro's patch next  
Issue: WICKET-3433  
  
Project: wicket.txt  
Commit: 3207346f424e76ead409676ff644127d31892574  
ClassName: org.apache.wicket.markup.parser.TagStack.java  
  
**+ private void assertOpenTagFor(ComponentTag closeTag) throws ParseException**+ {  
+ if (stack.size() > 0)  
+ {  
+ ComponentTag top = stack.pop();  
+ boolean mismatch = !hasEqualTagName(top, closeTag);  
+ if (mismatch)  
+ {  
+ top.setHasNoCloseTag(true);  
+ while (mismatch && !requiresCloseTag(top.getName()))  
+ {  
+ top.setHasNoCloseTag(true);  
+ if (stack.isEmpty())  
+ {  
+ break;  
+ }  
+ top = stack.pop();  
+ mismatch = !hasEqualTagName(top, closeTag);  
+ }  
+ if (mismatch)  
+ {  
+ throw new ParseException("Tag " + top.toUserDebugString() +  
+ " has a mismatched close tag at " + closeTag.toUserDebugString(),  
+ top.getPos());  
+ }  
+ }  
+ closeTag.setOpenTag(top);  
+ }  
+ else  
+ {  
+ throw new WicketParseException("Tag does not have a matching open tag:", closeTag);  
+ }  
+ }  
========================================

DifUtil.validatePage has an extra boolean param now so that it will  
**will call junit.assert if that is true.**Now you see in eclipse directly the wrong string in the stacktrace  
and you have support for the compare dialog that the junit view has  
also some junit fixes (form has now always an counter added)  
  
Project: wicket.txt  
Commit: 5c61480e82165eacf78f4435fe9fca4ca16384e4  
ClassName: wicket.util.diff.DiffUtil.java  
  
- public static final boolean validatePage(String document, final Class clazz, final String file)  
**+ public static final boolean validatePage(String document, final Class clazz, final String file, boolean failWithAssert)** throws IOException  
========================================

fixed package typo  
  
Project: wicket.txt  
Commit: 040d272f27b963b61de30f4980219141a6c2ed7f  
ClassName: wicket.util.instrument.SizeOfAgent.java  
  
+ private static long internalSizeOf(Object obj, Stack<Object> stack,  
+ Map<Object, Object> visited) {  
+ if (skipObject(obj, visited)) {  
+ return 0;  
+ }  
+ visited.put(obj, null);  
+ long result = 0;  
+ result += SizeOfAgent.sizeOf(obj);  
+ Class clazz = obj.getClass();  
+ if (clazz.isArray()) {  
+ if (clazz.getName().length() != 2) {// skip primitive type array  
+ int length = Array.getLength(obj);  
+ for (int i = 0; i < length; i++) {  
+ stack.add(Array.get(obj, i));  
+ }  
+ }  
+ return result;  
+ }  
+ while (clazz != null) {  
+ Field[] fields = clazz.getDeclaredFields();  
+ for (int i = 0; i < fields.length; i++) {  
+ if (!Modifier.isStatic(fields[i].getModifiers())) {  
+ if (fields[i].getType().isPrimitive()) {  
+ continue; // skip primitive fields  
+ } else {  
+ fields[i].setAccessible(true);  
+ try {  
+ Object objectToAdd = fields[i].get(obj);  
+ if (objectToAdd != null) {  
+ stack.add(objectToAdd);  
+ }  
+ } catch (IllegalAccessException ex) {  
**+ assert false;**+ }  
+ }  
+ }  
+ }  
+ clazz = clazz.getSuperclass();  
+ }  
+ return result;  
+ }  
========================================

Last fixes I hope for parent restructure  
  
Project: wicket.txt  
Commit: 4f0742c057df321c2313a9acb59a67a4ace13be2  
ClassName: wicket.util.license.PropertiesLicenseHeaderHandler.java  
  
+ public boolean checkLicenseHeader(File file)  
+ {  
+ Revision revision = null;  
+ try  
+ {  
+ String header = extractLicenseHeader(file, 0, 14);  
+ revision = Diff.diff(getLicenseHeader().split(LINE\_ENDING), header.split(LINE\_ENDING));  
+ }  
+ catch (Exception e)  
+ {  
**+ Assert.fail(e.getMessage());**+ }  
+ return revision.size() == 0;  
+ }  
========================================

WICKET-3008 fixed: clarify purpose of getMarkupSettings().setThrowExceptionOnMissingXmlDeclaration(true)  
re-enabled ApacheLicenseHeaderTest and fixed the identified issues  
  
Project: wicket.txt  
Commit: 4a57bf8d36dd02cf94dc6b3ae109522659a1c3fe  
ClassName: org.apache.wicket.util.license.XmlPrologHeaderHandler.java  
  
+ public boolean checkLicenseHeader(final File file)  
+ {  
+ Revision revision = null;  
+ try  
+ {  
+ String header = extractLicenseHeader(file, 0, 1);  
+ return header.startsWith("<?xml") == true;  
+ }  
+ catch (Exception e)  
+ {  
+ e.printStackTrace();  
**+ Assert.fail(e.getMessage());**+ }  
+ return true;  
+ }  
========================================

Last fixes I hope for parent restructure  
  
Project: wicket.txt  
Commit: 4f0742c057df321c2313a9acb59a67a4ace13be2  
ClassName: wicket.util.diff.DiffUtil.java  
  
+ public static final boolean validatePage(String document, final Class clazz, final String file,  
**+ boolean failWithAssert) throws IOException**+ {  
+ String filename = clazz.getPackage().getName();  
+ filename = filename.replace('.', '/');  
+ filename += "/" + file;  
+ InputStream in = clazz.getClassLoader().getResourceAsStream(filename);  
+ if (in == null)  
+ {  
+ throw new IOException("File not found: " + filename);  
+ }  
+ String reference = Streams.readString(in);  
+ reference = reference.replaceAll("\n\r", "\n");  
+ reference = reference.replaceAll("\r\n", "\n");  
+ document = document.replaceAll("\n\r", "\n");  
+ document = document.replaceAll("\r\n", "\n");  
+ boolean equals = document.equals(reference);  
+ if (equals == false)  
+ {  
+ if (Boolean.getBoolean("wicket.replace.expected.results"))  
+ {  
+ in.close();  
+ in = null;  
+ replaceExpectedResultFile(document, clazz, file);  
+ return true;  
+ }  
**+ if (failWithAssert)**+ {  
**+ Assert.assertEquals(filename, reference, document);**+ }  
+ else  
+ {  
+ log.error("File name: " + file);  
+ log.error("===================");  
+ log.error(reference);  
+ log.error("===================");  
+ log.error(document);  
+ log.error("===================");  
+ String[] test1 = StringList.tokenize(reference, "\n").toArray();  
+ String[] test2 = StringList.tokenize(document, "\n").toArray();  
+ Diff df = new Diff(test1);  
+ Revision r;  
+ try  
+ {  
+ r = df.diff(test2);  
+ }  
+ catch (DifferentiationFailedException e)  
+ {  
+ throw new RuntimeException(e);  
+ }  
+ System.out.println(r.toString());  
+ }  
+ }  
+ return equals;  
+ }  
========================================

Last fixes I hope for parent restructure  
  
Project: wicket.txt  
Commit: 4f0742c057df321c2313a9acb59a67a4ace13be2  
ClassName: wicket.util.license.VelocityLicenseHeaderHandler.java  
  
+ public boolean checkLicenseHeader(File file)  
+ {  
+ Revision revision = null;  
+ try  
+ {  
+ String header = extractLicenseHeader(file, 0, 16);  
+ revision = Diff.diff(getLicenseHeader().split(LINE\_ENDING), header.split(LINE\_ENDING));  
+ }  
+ catch (Exception e)  
+ {  
**+ Assert.fail(e.getMessage());**+ }  
+ return revision.size() == 0;  
+ }  
========================================

RESOLVED - issue WICKET-1575: AjaxEventBehavior does not check for component.isEnableAllowed before adding the ajax event to the tag  
https://issues.apache.org/jira/browse/WICKET-1575  
  
Project: wicket.txt  
Commit: c01cbf5de53555db7878ee9ca142d3e102177ec4  
ClassName: org.apache.wicket.ajax.AjaxBehaviorEnabledPage.java  
  
+ protected void onEvent(AjaxRequestTarget target)  
+ {  
**+ Assert.fail("should never get here with disabled ajax");**+ }  
========================================

Making sure that client URL is context relative even while responding errors  
Issue: WICKET-4168  
  
Project: wicket.txt  
Commit: 5ad040a3c22a5ad92dc8c7d6e0333b59c6a01e5d  
ClassName: org.apache.wicket.protocol.http.servlet.ServletWebRequest.java  
  
- private Url getUrl(HttpServletRequest request, String filterPrefix)  
- {  
- if (filterPrefix.length() > 0 && !filterPrefix.endsWith("/"))  
- {  
- filterPrefix += "/";  
- }  
- StringBuilder url = new StringBuilder();  
- String uri = request.getRequestURI();  
- uri = Strings.stripJSessionId(uri);  
- final int start = request.getContextPath().length() + filterPrefix.length() + 1;  
- url.append(uri.substring(start));  
+ Url errorClientUrl = errorWebRequest.getClientUrl();  
- if (errorAttributes == null)  
- {  
- String query = request.getQueryString();  
- if (!Strings.isEmpty(query))  
- {  
- url.append('?');  
- url.append(query);  
- }  
- }  
**+ assertEquals("any/source/of/error", errorClientUrl.toString());**- return setParameters(Url.parse(url.toString(), getCharset()));  
 }  
========================================

fixed: Form#anyComponentError change in 1.4 breaks validation. Couldn't really test it since 1.5 doesn't yet compile.  
Issue: WICKET-2374  
  
Project: wicket.txt  
Commit: 73bcfb287a4f4161a2ecec51ac1cf2f3cc603736  
ClassName: org.apache.wicket.markup.html.form.FormHasErrorPage.java  
  
+ protected void onSubmit()  
+ {  
+ super.onSubmit();  
+ labelModel.setObject("Test FAILED - an error was expected");  
**+ Assert.fail("A validation error should've been detected by the Form processing");**+ }  
========================================

[WSS-191] - Moved trust verification from WSHandler to SignatureProcessor  
- Trust is now verified on certificates \*before\* signature validation  
- This means that bad requests are rejected quicker  
- Also it works around a security bug in the WSHandler implementations where trust is only verified on a single certificate  
- Also did a general rewrite of the SignatureProcessor  
  
Project: wss4j.txt  
Commit: 7e036c7397e5703b91ef5885bee37d238d7ca0c8  
ClassName: org.apache.ws.security.processor.SignatureProcessor.java  
  
+ private static Principal createPrincipalFromSAMLKeyInfo(  
+ SAMLKeyInfo samlKeyInfo  
+ ) {  
+ X509Certificate[] samlCerts = samlKeyInfo.getCerts();  
+ Principal principal = null;  
+ if (samlCerts != null && samlCerts.length > 0) {  
+ principal = samlCerts[0].getSubjectX500Principal();  
+ } else {  
**+ final SAMLAssertion assertion = samlKeyInfo.getAssertion();  
+ principal = new CustomTokenPrincipal(assertion.getId());  
+ ((CustomTokenPrincipal)principal).setTokenObject(assertion);**+ }  
+ return principal;  
+ }  
========================================

[WSS-178] - A fix for "signature verification failure of signed saml token"  
**- The problem was that the SAML Assertion was below the STR, which assumed the Assertion would always have been previously processed**- Also fixed the "No message with ID..." exception that was being thrown in XML Security.  
  
Project: wss4j.txt  
Commit: 632b98e88469a23535a24478899dabcea96af2ff  
ClassName: org.apache.ws.security.util.WSSecurityUtil.java  
  
**+ public static Element findSAMLAssertionElementById(Node startNode, String value) {**+ Element foundElement = null;  
+ if (startNode == null) {  
+ return null;  
+ }  
========================================

[WSS-178] - A fix for "signature verification failure of signed saml token"  
**- The problem was that the SAML Assertion was below the STR, which assumed the Assertion would always have been previously processed**- Also fixed the "No message with ID..." exception that was being thrown in XML Security.  
  
Project: wss4j.txt  
Commit: 632b98e88469a23535a24478899dabcea96af2ff  
ClassName: org.apache.ws.security.message.token.SecurityTokenReference.java  
  
+ private Element findTokenElement(  
+ Document doc,  
+ WSDocInfo docInfo,  
+ CallbackHandler cb,  
+ String uri,  
+ String type  
+ ) {  
 Element tokElement = null;  
**- String saml10 = WSConstants.WSS\_SAML\_NS + WSConstants.SAML\_ASSERTION\_ID;**- if (saml10.equals(type)  
- || WSConstants.WSC\_SCT.equals(type)) {  
+ String id = uri;  
+ if (id.charAt(0) == '#') {  
+ id = id.substring(1);  
+ }  
**+ String assertionStr = WSConstants.WSS\_SAML\_NS + WSConstants.ASSERTION\_LN;**+ if (WSConstants.WSS\_SAML\_KI\_VALUE\_TYPE.equals(type)  
**+ || assertionStr.equals(type)) {  
 Element sa = docInfo.getAssertion();**- String saID = null;  
 if (sa != null) {  
**- saID = sa.getAttribute("AssertionID");**- }  
**+ String saID = sa.getAttribute("AssertionID");** if (doDebug) {  
========================================