Item 85
git_comments:

1. If the ParseDateField URP isn't ahead of the DUP, then the date won't be normalized in the buffered tlog entry, and the doc won't be indexed on the replaying replica - a warning is logged as follows: WARN [...] o.a.s.u.UpdateLog REYPLAY_ERR: IOException reading log org.apache.solr.common.SolrException: Invalid Date String:'2017-01-05' at org.apache.solr.util.DateMathParser.parseMath(DateMathParser.java:234) at org.apache.solr.schema.TrieField.createField(TrieField.java:725) [...]
2. means that we've seen the leader and have version info (i.e. we are a non-leader replica)
3. Invalid date will be normalized by ParseDateField URP
4. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
5. Non-JSON types (Date in this case) aren't handled properly in noggit-0.6. Although this is fixed in https://github.com/yonik/noggit/commit/ec3e732af7c9425e8f40297463cbe294154682b1 to call obj.toString(), Date::toString produces a Date representation that Solr doesn't like, so we convert using Instant::toString

git_commits:

1. **summary:** SOLR-9883: In example schemaless configs' default update chain, move the DUP to after the AddSchemaFields URP (which is now tagged as RunAlways), to avoid invalid buffered tlog entry replays.
   message: SOLR-9883: In example schemaless configs' default update chain, move the DUP to after the AddSchemaFields URP (which is now tagged as RunAlways), to avoid invalid buffered tlog entry replays.

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** example solr config files can lead to invalid tlog replays when using add-unknown-fields-to-schema updat chain
   description: The current basic_configs and data_driven_schema_configs try to create unknown fields. The problem is that the date processing "ParseDateFieldUpdateProcessorFactory" is not invoked if the doc is replayed from the tlog. Whether there are other places this is a problem I don't know, this is a concrete example that fails in the field. So say I have a pattern for dates that omits the trialing 'Z', as: yyyy-MM-dd'T'HH:mm:ss.SSS This work fine when the doc is initially indexed. Now say the doc must be replayed from the tlog. The doc errors out with "unknown date format" since (apparently) this doesn't go through the same update chain, perhaps due to the sample configs defining ParseDateFieldUpdateProcessorFactory after DistributedUpdateProcessorFactory?
2. **summary:** example solr config files can lead to invalid tlog replays when using add-unknown-fields-to-schema updat chain
   description: The current basic_configs and data_driven_schema_configs try to create unknown fields. The problem is that the date processing "ParseDateFieldUpdateProcessorFactory" is not invoked if the

doc is replayed from the tlog. Whether there are other places this is a problem I don't know, this is a concrete example that fails in the field. So say I have a pattern for dates that omits the trialing 'Z', as: yyyy-MM-dd'T'HH:mm:ss.SSS This work fine when the doc is initially indexed. Now say the doc must be replayed from the tlog. The doc errors out with "unknown date format" since (apparently) this doesn't go through the same update chain, perhaps due to the sample configs defining ParseDateFieldUpdateProcessorFactory after DistributedUpdateProcessorFactory?

3. **summary:** example solr config files can lead to invalid tlog replays when using add-unknown-fields-to-schema updat chain
   description: The current basic_configs and data_driven_schema_configs try to create unknown fields. The problem is that the date processing "ParseDateFieldUpdateProcessorFactory" is not invoked if the doc is replayed from the tlog. Whether there are other places this is a problem I don't know, this is a concrete example that fails in the field. So say I have a pattern for dates that omits the trialing 'Z', as: yyyy-MM-dd'T'HH:mm:ss.SSS This work fine when the doc is initially indexed. Now say the doc must be replayed from the tlog. The doc errors out with "unknown date format" since (apparently) this doesn't go through the same update chain, perhaps due to the sample configs defining ParseDateFieldUpdateProcessorFactory after DistributedUpdateProcessorFactory?
4. **summary:** example solr config files can lead to invalid tlog replays when using add-unknown-fields-to-schema updat chain
   description: The current basic_configs and data_driven_schema_configs try to create unknown fields. The problem is that the date processing "ParseDateFieldUpdateProcessorFactory" is not invoked if the doc is replayed from the tlog. Whether there are other places this is a problem I don't know, this is a concrete example that fails in the field. So say I have a pattern for dates that omits the trialing 'Z', as: yyyy-MM-dd'T'HH:mm:ss.SSS This work fine when the doc is initially indexed. Now say the doc must be replayed from the tlog. The doc errors out with "unknown date format" since (apparently) this doesn't go through the same update chain, perhaps due to the sample configs defining ParseDateFieldUpdateProcessorFactory after DistributedUpdateProcessorFactory?
   label: documentation
5. **summary:** example solr config files can lead to invalid tlog replays when using add-unknown-fields-to-schema updat chain
   description: The current basic_configs and data_driven_schema_configs try to create unknown fields. The problem is that the date processing "ParseDateFieldUpdateProcessorFactory" is not invoked if the doc is replayed from the tlog. Whether there are other places this is a problem I don't know, this is a concrete example that fails in the field. So say I have a pattern for dates that omits the trialing 'Z', as: yyyy-MM-dd'T'HH:mm:ss.SSS This work fine when the doc is initially indexed. Now say the doc must be replayed from the tlog. The doc errors out with "unknown date format" since (apparently) this doesn't go through the same update chain, perhaps due to the sample configs defining ParseDateFieldUpdateProcessorFactory after DistributedUpdateProcessorFactory?
6. **summary:** example solr config files can lead to invalid tlog replays when using add-unknown-fields-to-schema updat chain
   description: The current basic_configs and data_driven_schema_configs try to create unknown fields. The problem is that the date processing "ParseDateFieldUpdateProcessorFactory" is not invoked if the doc is replayed from the tlog. Whether there are other places this is a problem I don't know, this is a concrete example that fails in the field. So say I have a pattern for dates that omits the trialing 'Z', as: yyyy-MM-dd'T'HH:mm:ss.SSS This work fine when the doc is initially indexed. Now say the doc must be replayed from the tlog. The doc errors out with "unknown date format" since (apparently) this doesn't go through the same update chain, perhaps due to the sample configs defining ParseDateFieldUpdateProcessorFactory after DistributedUpdateProcessorFactory?
7. **summary:** example solr config files can lead to invalid tlog replays when using add-unknown-fields-to-schema updat chain
   description: The current basic_configs and data_driven_schema_configs try to create unknown fields. The problem is that the date processing "ParseDateFieldUpdateProcessorFactory" is not invoked if the doc is replayed from the tlog. Whether there are other places this is a problem I don't know, this is a concrete example that fails in the field. So say I have a pattern for dates that omits the trialing 'Z', as: yyyy-MM-dd'T'HH:mm:ss.SSS This work fine when the doc is initially indexed. Now say the doc must be replayed from the tlog. The doc errors out with "unknown date format" since (apparently) this doesn't go through the same update chain, perhaps due to the sample configs defining ParseDateFieldUpdateProcessorFactory after DistributedUpdateProcessorFactory?
8. **summary:** example solr config files can lead to invalid tlog replays when using add-unknown-fields-to-schema updat chain

description: The current basic_configs and data_driven_schema_configs try to create unknown fields. The problem is that the date processing "ParseDateFieldUpdateProcessorFactory" is not invoked if the doc is replayed from the tlog. Whether there are other places this is a problem I don't know, this is a concrete example that fails in the field. So say I have a pattern for dates that omits the trialing 'Z', as: yyyy-MM-dd'T'HH:mm:ss.SSS This work fine when the doc is initially indexed. Now say the doc must be replayed from the tlog. The doc errors out with "unknown date format" since (apparently) this doesn't go through the same update chain, perhaps due to the sample configs defining ParseDateFieldUpdateProcessorFactory after DistributedUpdateProcessorFactory?
label: documentation

9. **summary:** example solr config files can lead to invalid tlog replays when using add-unknown-fields-to-schema updat chain
description: The current basic_configs and data_driven_schema_configs try to create unknown fields. The problem is that the date processing "ParseDateFieldUpdateProcessorFactory" is not invoked if the doc is replayed from the tlog. Whether there are other places this is a problem I don't know, this is a concrete example that fails in the field. So say I have a pattern for dates that omits the trialing 'Z', as: yyyy-MM-dd'T'HH:mm:ss.SSS This work fine when the doc is initially indexed. Now say the doc must be replayed from the tlog. The doc errors out with "unknown date format" since (apparently) this doesn't go through the same update chain, perhaps due to the sample configs defining ParseDateFieldUpdateProcessorFactory after DistributedUpdateProcessorFactory?

jira_issues_comments:

1. There's quite a bit of discussion at SOLR-8030 that's relevant. I don't quite know whether the simple expedient of putting the URPs before the DistribUpdateProcessorFactory is sufficient (or safe).
2. Attaching a patch that switches example configs's add-unknown-fields-to-schema update chains so that the DUP is after the AddSchemaFields URPF. In my manual testing (see below), this prevents the data corruption: the buffered tlog entry includes the date normalization. I also made {{AddSchemaFields}} URPF implement {{UpdateRequestProcessorFactory.RunAlways}}, so that schema modifications will continue to be applied on all replicas (the original rationale for moving the DUP position on SOLR-6137). Following an offline reproduction suggestion from [~hossman], I was able to manually reproduce the data corruption as follows: # Added an artificial 1-minute delay in {{PeerSync}} # {{bin/solr start -e cloud # nodes=2, coll=gettingstarted, shards=1, rf=2, configset=data_driven_schema_configs}} # {{curl -X POST -H 'Content-type: application/xml' http://localhost:8983/solr/gettingstarted/update -d '<add> <doc><field name="f_dt">2015-06-09</field></doc></add>'}} # {{kill -9 $(cat bin/solr-7574.pid)}} # {{curl -X POST -H 'Content-type: application/xml' http://localhost:8983/solr/gettingstarted/update -d '<add><doc><field name="f_dt">2015-06-10</field></doc></add>'}} # {{bin/solr start -cloud -p 7574 -s "example/cloud/node2/solr" -z localhost:9983}} # {{curl -X POST -H 'Content-type: application/xml' http://localhost:8983/solr/gettingstarted/update -d '<add><doc><field name="f_dt">2015-06-11</field> </doc></add>'}} I had to add step #3 to create a transaction log entry on the 7574 replica prior to shutdown; otherwise on restart it would refuse to perform peer sync, because it didn't know where to start (due to no recent versions in the tlog) and instead initiated full recovery. I'm working on an automated data corruption test. I want to get this change into the 6.4 release.
3. Forgot to mention: with the attached patch, I was no longer able to reproduce the data corruption with the above method.
4. **body:** Patch with a new automated data corruption test. I tried to make a cloud test, but I couldn't get it to work. Instead, the test in the patch simulates this situation by directly turning on tlog buffering mode in a single core, and sending in an update (with param {{update.distrib=fromleader}}) after manually running the "add-unknown-fields-to-schema" update chain on it up through the DUP. The test succeeds with the solr config modifications in the patch, and fails without it. The patch also fixes a typos in the replay failure log message ({{REYPLAY}}->{{REPLAY}}). I'm running all Solr tests and precommit now. When they succeed, I'll commit.
label: documentation
5. Updated patch, moves config files to temp dir to avoid permission failures when auto-upgrading the schema file to {{managed-schema}}. (Didn't see this failure when running from IntelliJ.) All Solr tests pass, and precommit passes. Committing shortly.
6. Commit 9a6ff177b6f7c776cc6bf4625ed2d5dd7cce81d2 in lucene-solr's branch refs/heads/branch_6x from [~steve_rowe] [https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=9a6ff17] SOLR-9883: In example schemaless configs' default update chain, move the DUP to after the AddSchemaFields URP (which is now tagged as RunAlways), to avoid invalid buffered tlog entry replays.

7. Commit d817fd43eccd67a5d73c3bbc49561de65d3fc9cb in lucene-solr's branch refs/heads/master from [~steve_rowe] [https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=d817fd4] SOLR-9883: In example schemaless configs' default update chain, move the DUP to after the AddSchemaFields URP (which is now tagged as RunAlways), to avoid invalid buffered tlog entry replays.