

Item 78

git_comments:

git_commits:

1. **summary:** LUCENE-3520: fix test failure: call SIS.getVersion directly to avoid possible ACE if we are closed while being reopened
message: LUCENE-3520: fix test failure: call SIS.getVersion directly to avoid possible ACE if we are closed while being reopened git-svn-id:
https://svn.apache.org/repos/asf/lucene/dev/branches/branch_3x@118495513f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** If the NRT reader hasn't changed then IndexReader.openIfChanged should return null
description: I hit a failure in TestSearcherManager (NOTE: doesn't always fail): {noformat} ant test -Dtestcase=TestSearcherManager -Dtestmethod=testSearcherManager -Dtests.seed=459ac99a4256789c;-29b8a7f52497c3b4:145ae632ae9e1ecf {noformat} It was tripping the assert inside SearcherLifetimeManager.record, because two different IndexSearcher instances had different IR instances sharing the same version. This was happening because IW.getReader always returns a new reader even when there are no changes. I think we should fix that... Separately I found a deadlock in TestSearcherManager.testIntermediateClose, if the test gets SerialMergeScheduler and needs to merge during the second commit.

jira_issues_comments:

1. Patch.
2. mike, I just had a quick look but is this intentionally? {code} - return IndexReader.openIfChanged(oldReader, writer, applyDeletes); + return IndexReader.openIfChanged(oldReader, applyDeletes); {code} seems like you are not opening a NRT reader there anymore?
3. Thanks Simon. It should still be opening an NRT reader: if you have an NRT reader (which we do here) and pass that to IR.openIfChanged, you'll always get back a new NRT reader (this is the contract of IR.openIfChanged). So I believe passing writer here wasn't necessary?
4. Steve just took a stack trace (and aborted the build) of a test run started yesterday. I noticed "SearchManager" in one of the traces. Maybe this issue fixes? <https://builds.apache.org/job/Lucene-Solr-tests-only-trunk/10840/console>
5. **body:** bq.It should still be opening an NRT reader: if you have an NRT reader (which we do here) and pass that to IR.openIfChanged, you'll always get back a new NRT reader (this is the contract of IR.openIfChanged). hmm, however the signature of openIfChanged(IR, boolean) actually refers to openIfChanged(IndexReader oldReader, boolean readonly) which seems confusing when you pass applyDeletes to it, no?
label: code-design
6. bq. I noticed "SearchManager" in one of the traces. Maybe this issue fixes? Indeed I think this fixes it; I just committed that test-only fix.
7. **body:** bq. hmm, however the signature of openIfChanged(IR, boolean) actually refers to openIfChanged(IndexReader oldReader, boolean readonly) which seems confusing when you pass applyDeletes to it, no? Ugh, you're right! In fact we don't need to pass applyDeletes either; this too is

inherited from the prior reader. So I'll just reduce it to `IR.openIfChanged(oldReader)`. Hmm then we can simplify `SearcherManager` some. I'll work out a new patch.

label: code-design

8. **body:** New patch; I was able to simplify `SearcherManager` since both cases (open-from-writer (= NRT case) and open-from-dir (= non-NRT case)) now call the same `IR.openIfChanged(oldReader)`.

label: code-design

9. bq. New patch; I was able to simplify `SearcherManager` since both cases (open-from-writer (= NRT case) and open-from-dir (= non-NRT case)) now call the same `IR.openIfChanged(oldReader)`. yeah nice! looks good mike!

10. Bulk close after release of 3.5