

CN Index - Task #7771

Story # 7769 (Closed): Improve the performance on solr index

Use multiple threads to index objects

2016-05-04 23:15 - Jing Tao

Status:	Closed	Start date:	2016-05-04
Priority:	Normal	Due date:	
Assignee:	Jing Tao	% Done:	100%
Category:	d1_cn_index_processor	Estimated time:	0.00 hour
Target version:	CCI-2.3.0		
Story Points:			

Description

Worked on generating index on multiple threads:

```
while (getNextTask() != null ) {  
    process(nextTask); }  
  
while (getNextTask() != null ) {  
    executor.submit(nextTask);  
}
```

Fortunately there is no shared class variables can't be static. So we don't need to lock them.

Handling resource maps has a race issue:

R1: s1 documents d1

R2: s1 documents d2

At the beginning, there is no documents and resourceMap on the solr index of s1.

Sequence:

After processing R1 and the solr index of s1:

documents d1

resourceMap R1

After processing R2 and the solr index of s1:

documents d1

documents d2

resourceMap R1

resourceMap R2

Concurrent:

1. Both threads to handle R1 and R2 read a copy without documents and resourceMap information.

2. Thread 1 handling R1 finished first and send it to the solr server:

documents d1

resourceMap R1

3. Thread 2 handling R2 finished later and send it to the solr server. It will overwrite what thread 1 did. So the eventual result will be:

documents d2

resourceMap R2

Wrong!

Handle resource map objects sequentially? no.

Proposed Solution:

1. Maintain a set containing the relevant objects' id (s1 and d1) when it processes a resource map
2. Before we process a resource map, check its relevant ids are on the set. If they are on the set, please wait and try again later (with max attempts); otherwise, put those ids on the set and start to process it.
3. The processing is done, remove those ids from the set

ConcurrentSkipListSet vs HashSet + lock vs Hash+ synchronize

Related issues:

Related to Infrastructure - Story #8172: investigate atomic updates for some ...	In Progress	2017-09-01
Related to Infrastructure - Story #8173: add checks for retrograde systemMeta...	New	2017-09-01

History

#1 - 2016-05-25 16:31 - Jing Tao

- Status changed from New to In Progress
- Category set to d1_cn_index_processor
- % Done changed from 0 to 30

#2 - 2016-12-21 09:15 - Dave Vieglais

- % Done changed from 30 to 100
- Status changed from In Progress to Closed

#3 - 2017-09-01 19:36 - Rob Nahf

- Related to Story #8172: investigate atomic updates for some solr updates added

#4 - 2017-09-01 19:42 - Rob Nahf

- Related to Story #8173: add checks for retrograde systemMetadata changes added