Item 1

git_comments:

git_commits:

1. **summary:** do not recalculate cache capacity post-compaction if it's been manually modified. patch by jbellis; reviewed by Ryan King for CASSANDRA-1079
   message: do not recalculate cache capacity post-compaction if it's been manually modified. patch by jbellis; reviewed by Ryan King for CASSANDRA-1079 git-svn-id: https://svn.apache.org/repos/asf/cassandra/branches/cassandra-0.6@945315 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Cache capacity settings done via nodetool get reset on memtable flushes
   description: In an experiment we set cache capacities via nodetool. The config file had the KeyCache for this CF at 1000000, we set the RowCache to 10000000 via nodetool. The next time we flushed a memtable for that CF, the cache capacity settings got reverted to what is in the conf file. We repeated the experiment with the same results.
2. **summary:** Cache capacity settings done via nodetool get reset on memtable flushes
   description: In an experiment we set cache capacities via nodetool. The config file had the KeyCache for this CF at 1000000, we set the RowCache to 10000000 via nodetool. The next time we flushed a memtable for that CF, the cache capacity settings got reverted to what is in the conf file. We repeated the experiment with the same results.
   label: code-design
3. **summary:** Cache capacity settings done via nodetool get reset on memtable flushes
   description: In an experiment we set cache capacities via nodetool. The config file had the KeyCache for this CF at 1000000, we set the RowCache to 10000000 via nodetool. The next time we flushed a memtable for that CF, the cache capacity settings got reverted to what is in the conf file. We repeated the experiment with the same results.
   label: code-design
4. **summary:** Cache capacity settings done via nodetool get reset on memtable flushes
   description: In an experiment we set cache capacities via nodetool. The config file had the KeyCache for this CF at 1000000, we set the RowCache to 10000000 via nodetool. The next time we flushed a memtable for that CF, the cache capacity settings got reverted to what is in the conf file. We repeated the experiment with the same results.
   label: code-design
5. **summary:** Cache capacity settings done via nodetool get reset on memtable flushes
   description: In an experiment we set cache capacities via nodetool. The config file had the KeyCache for this CF at 1000000, we set the RowCache to 10000000 via nodetool. The next time we flushed a memtable for that CF, the cache capacity settings got reverted to what is in the conf file. We repeated the experiment with the same results.
6. **summary:** Cache capacity settings done via nodetool get reset on memtable flushes
   description: In an experiment we set cache capacities via nodetool. The config file had the KeyCache for this CF at 1000000, we set the RowCache to 10000000 via nodetool. The next time we flushed a memtable for that CF, the cache capacity settings got reverted to what is in the conf file. We repeated the experiment with the same results.
7. **summary:** Cache capacity settings done via nodetool get reset on memtable flushes

description: In an experiment we set cache capacities via nodetool. The config file had the KeyCache for this CF at 1000000, we set the RowCache to 10000000 via nodetool. The next time we flushed a memtable for that CF, the cache capacity settings got reverted to what is in the conf file. We repeated the experiment with the same results.

8. **summary:** Cache capacity settings done via nodetool get reset on memtable flushes
   description: In an experiment we set cache capacities via nodetool. The config file had the KeyCache for this CF at 1000000, we set the RowCache to 10000000 via nodetool. The next time we flushed a memtable for that CF, the cache capacity settings got reverted to what is in the conf file. We repeated the experiment with the same results.
9. **summary:** Cache capacity settings done via nodetool get reset on memtable flushes
   description: In an experiment we set cache capacities via nodetool. The config file had the KeyCache for this CF at 1000000, we set the RowCache to 10000000 via nodetool. The next time we flushed a memtable for that CF, the cache capacity settings got reverted to what is in the conf file. We repeated the experiment with the same results.
10. **summary:** Cache capacity settings done via nodetool get reset on memtable flushes
    description: In an experiment we set cache capacities via nodetool. The config file had the KeyCache for this CF at 1000000, we set the RowCache to 10000000 via nodetool. The next time we flushed a memtable for that CF, the cache capacity settings got reverted to what is in the conf file. We repeated the experiment with the same results.
11. **summary:** Cache capacity settings done via nodetool get reset on memtable flushes
    description: In an experiment we set cache capacities via nodetool. The config file had the KeyCache for this CF at 1000000, we set the RowCache to 10000000 via nodetool. The next time we flushed a memtable for that CF, the cache capacity settings got reverted to what is in the conf file. We repeated the experiment with the same results.
12. **summary:** Cache capacity settings done via nodetool get reset on memtable flushes
    description: In an experiment we set cache capacities via nodetool. The config file had the KeyCache for this CF at 1000000, we set the RowCache to 10000000 via nodetool. The next time we flushed a memtable for that CF, the cache capacity settings got reverted to what is in the conf file. We repeated the experiment with the same results.
13. **summary:** Cache capacity settings done via nodetool get reset on memtable flushes
    description: In an experiment we set cache capacities via nodetool. The config file had the KeyCache for this CF at 1000000, we set the RowCache to 10000000 via nodetool. The next time we flushed a memtable for that CF, the cache capacity settings got reverted to what is in the conf file. We repeated the experiment with the same results.

jira_issues_comments:

1. this is working as designed. the config file is The Source Of Truth for settings it contains.
2. **body:** If we're not going to let the jmx-based setting live longer than one memtable, we should probably remove the ability to set it that way. The current behavior is too surprising.
   label: code-design
3. **body:** It's not surprising if you remember that the conf is The Source Of Truth. :) I'm big on "there should only be one way to do it" but restarting to test cache size effects is too painful.
   label: code-design
4. **body:** ??I'm big on "there should only be one way to do it" but restarting to test cache size effects is too painful.?? I'm confused. Did you mean "isn't"?
   label: code-design
5. I think the disconnect is that Ryan and Jeff would like to be able to tune cache sizes on a cluster that is receiving writes, which is difficult when it is reset per-memtable.
6. Oh. Sometimes I need to read things twice -- yes, it's a bug if it's not persisting between flushes. (I'm saying that it's not supposed to persist b/t restarts, unless you change the conf file too.)
7. Yes, "not persisting between restarts" is expected, it currently gets reset when at every flush.
8. It appears the problem is that SSTableTracker's replace method calls updateCacheSizes, which reads the cache settings from DatabaseDescriptor (which doesn't get updated by the jmx command).
9. patch to not recalculate cache capacity if it's been manually modified
10. Patch looks good to me.
11. committed