

**git\_comments:**

1. node metrics
2. overseer metrics

**git\_commits:**

1. **summary:** SOLR-14965: add overseer queue size metrics (#2040)  
**message:** SOLR-14965: add overseer queue size metrics (#2040) Adds two metrics to the SolrCloud Overseer: solr\_metrics\_overseer\_stateUpdateQueueSize and solr\_metrics\_overseer\_collectionWorkQueueSize with corresponding entries in the the Prometheus exporter's default/stock configuration. Co-authored-by: Saatchi Bhalla <s.bhalla@salesforce.com>

**github\_issues:****github\_issues\_comments:****github\_pulls:**

1. **title:** SOLR-14965 add overseer queue size metrics  
**body:** <!--\_(If you are a project committer then you may remove some/all of the following template.)\_-->  
 Before creating a pull request, please file an issue in the ASF Jira system for Lucene or Solr: \*  
<https://issues.apache.org/jira/projects/LUCENE> \* <https://issues.apache.org/jira/projects/SOLR> You will need to create an account in Jira in order to create an issue. The title of the PR should reference the Jira issue number in the form: \* LUCENE-####: <short description of problem or changes> \* SOLR-####: <short description of problem or changes> LUCENE and SOLR must be fully capitalized. A short description helps people scanning pull requests for items they can work on. Properly referencing the issue in the title ensures that Jira is correctly updated with code review comments and commits. --> #  
 Description Please provide a short description of the changes you're making with this pull request. The Overseer work queues stored in ZK and abstracted by the Overseer can give us a good indication of the health of the cluster - if messages are taking too long to dequeue or the queue is growing too large, we know that the Overseer is overloaded and we are going to overwhelm the cluster. This work adds metrics to track the size of the Overseer queues (ClusterStateUpdate queue and Collections Work Queue). #  
 Solution Please provide a short description of the approach taken to implement your solution. Registered these two size metrics in a shared metrics registry for the Overseer. The Overseer shared metrics registry is only initialized upon reference so this shouldn't have any impact on metrics when run in non-cloud mode. Also updated the local solr-exporter-config.xml so that these metrics are exported and can be viewed locally in Grafana. #  
 Tests Please describe the tests you've developed or run to confirm this patch implements the feature or solves the problem. Local testing was accomplished by sending CreateCollection load and monitoring a local grafana dashboard which accessed the metrics through prometheus-exporter. As the size of these queues change extremely quickly, it was difficult to see changes in these metrics quickly enough--and if the requests were sent too quickly we'd see local solr hosts OOMing. However, I was able to verify that the metrics are updating as the size of the queues change, and in a real environment with more sustained load we should see these metrics more accurately represent the state of the Overseer. Testing confirmed that we don't see the Overseer metrics registry in non-cloud mode. **\*\*/admin/metrics response containing added metrics\*\***  **\*\*local testing sending CreateCollection request load and seeing metrics update in real time\*\*** ![image](https://user-images.githubusercontent.com/16807693/97367388-5d4a5d80-187f-11eb-8c45-58abb0b0e0cc.png) #  
 Checklist Please review the following and check all that apply: - [x] I have reviewed the guidelines for [How to Contribute] (<https://wiki.apache.org/solr/HowToContribute>) and my code conforms to the standards described there to the best of my ability. - [x] I have created a Jira issue and added the issue ID to my pull request title. - [x] I have given Solr maintainers [access](<https://help.github.com/en/articles/allowing-changes-to-a-pull-request-branch-created-from-a-fork>) to contribute to my PR branch. (optional but recommended) - [x] I have developed this patch against the `master` branch. - [x] I have run `./gradlew check`. - [ ] I have added tests for my changes. - [x] I have added documentation for the [Ref Guide] (<https://github.com/apache/lucene-solr/tree/master/solr/solr-ref-guide>) (for Solr changes only).

## github\_pulls\_comments:

1. @sigram Would you mind taking another look at this PR whenever you have a chance?
2. > Saatchi; could you please comment here what this change in the config is about? Did you blend two metrics into one, and if so why? Maybe could you share a snippet of the prometheus exporter's output so we know clearly what the output of this "looks like" Hey David, sorry for the delay in getting back to you. I looked at this again, and decided that the two overseer queue metrics should be exported as separate metrics (rather than grouping them together under a single exported metric). Each of these queues represent separate responsibilities of the Overseer, and this allows for more flexibility in the future to add more metrics for each queue and group those together per queue. Based on the current status of solr-exporter.xml this is how the exported metrics will look: { "name": "solr\_metrics\_overseer\_stateUpdateQueueSize", "type": "GAUGE", "help": "See following URL: <https://lucene.apache.org/solr/guide/metrics-reporting.html>", "label\_names": [], "label\_values": [], "value": 0 } { "name": "solr\_metrics\_overseer\_collectionWorkQueueSize", "type": "GAUGE", "help": "See following URL: <https://lucene.apache.org/solr/guide/metrics-reporting.html>", "label\_names": [], "label\_values": [], "value": 0 }
3. Looks good now Saatchi! I'll merge tomorrow with the following CHANGES.txt note under "Improvements": SOLR-14965: metrics: Adds two metrics to the SolrCloud Overseer: solr\_metrics\_overseer\_stateUpdateQueueSize and solr\_metrics\_overseer\_collectionWorkQueueSize with corresponding entries in the the Prometheus exporter's default/stock configuration. (Saatchi Bhalla, Megan Carey, Andrzej Bialecki, David Smiley) @sigram if you prefer to or if you have lingering concerns, let me know.
4. Thanks @dsmiley! Looks good to me.

## github\_pulls\_reviews:

1. You should never create instances of SolrMetricManager in your components. Always use the single instance constructed in `CoreContainer.getMetricManager()`. The only reason it kind of worked for you was that you made it a shared registry, which is created once per JVM - but as I explain below in this case a shared registry is incorrect and should not be used.
2. You should create a `SolrMetricsContext` in Overseer, and create a child context here, and then pass it as the first argument to `registerGauge`. This helps to properly unregister old gauges from the registry (because a new instance of `ClusterStateUpdater` is created each time this Overseer instance becomes the leader, and then it registers a new instance of gauge with the same name). Using a static string "overseer" defeats this mechanism, because it falsely indicates that the life-cycle of `ClusterStateUpdater` is the same as its parent object Overseer. Then in the `ClusterStateUpdater.close()` method you should unregister this gauge using the value of `solrMetricsContext.getTag()`. This prevents obscure reference leaks in the metrics registry.
3. Please use the context tag value here.
4. See above - this should never be done.
5. See above comments - you should create a child context here (from the one created in Overseer) and use it in this call, and then unregister this gauge using this context's tag value.
6. See above - use the tag value from the context.
7. Overseer registry CANNOT be a shared registry because it's possible to have multiple Overseers in a single JVM, especially in unit tests. This should be a regular per-CoreContainer registry, which is created within the scope of the instance available from CC.getMetricManager().
8. See above - this can't be a shared registry.
9. Thanks for the feedback, will update accordingly.
10. @sigram Thanks for the feedback! Is there documentation around adding metrics that covers things like this?
11. @sigram Updated the PR with those proposed changes. Whenever you're free to take another look I'd appreciate it!
12. Are there any tests that instantiate OverseerCollectionConfigSetProcessor? (Just want to make sure this is the only time this method is used, since we changed the signature)
13. Same question here - would be good to double-check that this is the only place we instantiate OverseerTaskProcessor
14. We can update this doc comment now, since we are using the CoreContainer registry
15. chose to leave it this way - although we're using MetricManager from CoreContainer i'm still registering these metrics under "solr.overseer."

16. Updated the one test that calls this thanks for the catch!

17. @megancarey unfortunately, there's not much documentation apart from the javadocs and user-level info in the RefGuide. We should create a doc in solr/dev-docs/metrics.

**jira\_issues:**

**jira\_issues\_comments:**