Reducing skipped searches in Splunk Enterprise

You are a Splunk Administrator and are finding that sometimes your searches are getting skipped, which can cause you some wasted time and frustration. You know that searches can be skipped because the load on the system is higher than the resources you have available, but even after optimizing your searches to the best of your ability, you’re still experiencing skips.

You suspect that you need to make changes to your Splunk configuration to reduce the amount of skipped searches, but you’re not sure where to start. There are a number of different reasons that searches can be skipped, and learning more about these reasons can allow you to make changes which prevent skips from occurring.

Search concurrency

Splunk restricts the number of concurrent searches running on the system, which you can think of as search slots. This is done to protect the system from slowing and stopping if the search workload is much higher than resources available.

By default, the system total maximum concurrency (maximum search slots) is calculated based on the number of CPU cores on a search head (SH) or across a search head cluster (SHC). Both scheduled searches and ad-hoc searches use these search slots. There is a default limit of the search slots that scheduled searches can use, but there is no default limit on ad-hoc searches.

This means that ad-hoc searches can use up to the maximum amount of search slots, essentially leaving none for scheduled searches. Some of these calculations are shown below for a single search head, but the principal can be applied to a search head cluster as well:

- Total maximum concurrency = \(<\text{max\_searches\_per\_cpu}> \times \text{number of CPU cores in SH/SHC} + \)<base\_max\_searches>
- Maximum concurrent scheduled searches = \(<\text{max\_searches\_perc}> \times \text{total max concurrency}\)
- Maximum auto-summarization searches = \(<\text{auto\_summary\_perc}> \times \text{max concurrent scheduled searches}\)

Default values are:

- max_searches_per_cpu=1
- base_max_searches=6
- auto_summary_perc=50%
- max_searches_perc=50%

As well as being aware of these calculations, you can use Ad hoc search quota control to limit ad-hoc searches so they do not take away all search slots.
Find and reduce your skipped searches

Reducing skipped searches depends on identifying the bottleneck in the system. Some bottlenecks are soft (software configuration related) and others are hard (system resource limits).

You can run this search to find some of the reasons that your searches are skipping:

```
| index=\_internal sourcetype=scheduler savedsearch_name=\* status=skipped | stats count BY reason
```

Here are some of the most common reasons that you'll see to explain why your searches are skipping:

**User or role quota limit has been reached**

If you have programmed user or role quotas, certain searches may skip if these quota limits are breached. This is the intended function of quotas - to limit the number of concurrent searches a user or users within a role can run concurrently. You can change user or role quotas to fix this.

Sometimes you will also see that searches from certain users are skipped because the user doesn't have permissions to run scheduled jobs. [Here](#) is a good resource on resolving this issue.

**The maximum number of concurrent running jobs for a historical scheduled search has been reached**

Scheduled searches run at a certain frequency (e.g. every 5 minutes) and by default only 1 instance of a scheduled search can run at any given time. This limit is defined by `<max_concurrent>` in the `savedsearches.conf` file and generally, you don’t need to change this attribute to more than 1 (default). If a scheduled search job cannot be completed before the next period starts, it will be skipped. For instance, if a job takes 10 minutes to complete and runs at a 5 minute frequency, then it will skip.

To fix this problem, you need to identify which scheduled searches are skipped due to this reason and then do one of the following:

- Change the schedule frequency so it is higher than execution time for the search.
- Or, try to reduce the search execution time:
  - If you are resource constrained, you can use [workload management](#) to put these searches in a high priority pool to get them more resources so they complete faster.
  - Check the load on indexers to see if they are resource constrained and you need to increase resources. You can run this search or look at the monitoring console to see CPU utilization on indexers:

```
| index=\_introspection host=idx\* component=Hostwide | eval total_cpu_usage = \'data.cpu_system_pct\' + \'data.cpu_user_pct\'
```

The information provided in Splunk Lantern is intended for informational and educational purposes only. All information is provided in good faith, however, Splunk disclaims any and all representations and warranties, express and implied, regarding the information provided, including without limitation any warranties and representations regarding the completeness, adequacy or accuracy of the information. You agree to take full responsibility for the results arising from the use of the information provided.
If the output looks like the image below, this is why searches have a long execution time. In this instance, you'll need to add more indexer resources.

```
| timechart span=10m perc95(total_cpu_usage) BY host
```

The maximum number of concurrent auto-summarization searches has been reached

The number of summarization searches that can run concurrently is capped and that limit is defined by `<auto_summary_perc>` in the `limits.conf` file. This attribute may be raised to 75% to allow auto-summarization searches to be a higher percentage of the overall scheduled search limit, reducing the skipped searches.

You can also check if you are seeing skipped searches at a particular time, such as the top of the hour or 30 minutes past the hour. This means you have scheduled too many searches at the same time. You can resolve this issue by allowing the scheduler to optimize the search dispatch time by setting `<allow_skew>` or `<schedule_window>`.

The maximum number of concurrent historical scheduled searches on an instance or cluster has been reached

This is the most common reason for skipped searches. This happens because the searches have taken all available search slots so new searches cannot be scheduled.

**Too many searches are scheduled at the same time**

Check if your search schedule aligns at specific times. If this is the issue, you can try to use the `<allow_skew>` or `<schedule_window>` settings.

**Ad-hoc searches are taking up too many search slots**

Ad-hoc searches can use all available search slots, leaving none for scheduled searches. To see if this problem is the cause of your skipped searches, find out if there is a pattern to skipping by using the search below, then correlate the periods of high skipped searches with the number of ad-hoc searches running. If this is an issue, you can limit ad-hoc searches to a percentage of total maximum concurrency using the Ad hoc search quota feature in admission rules.

```
| index=_internal sourcetype=scheduler savedsearch_name=* status=skipped | bin _time=1h | stats count by _time
```
Indexer resource utilization (P95) is higher than 80%

Most parts of a search run on indexers. If indexers are bottlenecked due to too many searches or badly written searches (which causes cache churn) or high ingest load, the searches may take longer time to complete, so they will hold the search slots for longer and new searches cannot be dispatched. To see if this problem is the cause of your skipped searches, find out if there is a pattern to skipping by using the search in section 4b. Then, correlate the periods of high skipped searches with the CPU utilization of indexers. The best option is to add more capacity to the indexer tier.

The below image shows an example of skipped searches correlating with indexer load.

The total maximum concurrency limit is too low

Often the default maximum concurrency limit is too low for a workload environment. Especially if you have short searches, you may want to raise this default limit by increasing `<max_searches_per_cpu>` in a range of 2-5 in steps. Before increasing this setting, check if the indexer and search head CPU utilization are low enough (P95 < 50-60%).

The maximum concurrency limit for scheduled searches is too low

The concurrent scheduled searches are limited to 50% of the total maximum concurrency by default. If you are not running too many concurrent ad-hoc searches, you may want to raise `<max_searches_perc>` from 50 to 75% or higher. This will not help if ad-hoc searches increase because they will overrun the scheduled searches unless you use the Ad hoc search quota feature to limit ad-hoc searches.

Next steps

The content in this guide comes from a previously published blog, one of the thousands of Splunk resources available to help users succeed. In addition, these Splunk resources might help you understand and implement this use case:

- Blog: Schedule windows vs. skewing