Splunk® Enterprise Knowledge Manager Manual 7.3.0

Overview of summary-based search acceleration

Generated: 7/21/2019 6:24 pm
Overview of summary-based search acceleration

Splunk Enterprise is capable of generating reports on massive amounts of data. However, the amount of time it takes to compute such reports is directly proportional to the number of events they summarize. Plainly put, it can take a lot of time to report on very large datasets. If you only have to do this on an occasional basis, the length of time may not be an issue. But running such reports on a regular schedule (or using them as the basis for panels in popular dashboards) is impractical—and this impracticality only increases exponentially as more and more users in your organization run similar reports.

To efficiently report on large volumes of data, you need to create data summaries that are populated by the results of background runs of the search upon which the report is based. When you next run the report against data that has been summarized in this manner, it should complete significantly faster, because the summaries are much smaller than the original events from which they were generated.

There are three data summary creation methods:

- **Report acceleration** - Uses automatically-created summaries to speed up completion times for certain kinds of reports.
- **Data model acceleration** - Uses automatically-created summaries to speed up completion times for pivot-based reports and dashboard panels.
- **Summary indexing** - Enables acceleration of searches and reports through the manual creation of separate summary indexes that exist separately from your main indexes.

**Report acceleration**

**Report acceleration** is used to accelerate individual reports. It's easy to set up for any **transforming search** or report that runs over a large dataset.

In early versions of Splunk software, summary indexing was used to accelerate reports. Report acceleration is preferable over summary indexing for the following reasons:

- **Kicking off report acceleration is as easy as clicking a checkbox and selecting a time range.** Everything after that happens behind the scenes. Subsequent runs of accelerated reports should complete faster as long as they're run (at least partially) within their selected time ranges. For summary indexing you need to design a search to populate the index that
includes special search commands; you may need to create the summary index as well.

• **Splunk software automatically shares report acceleration summaries with similar searches.** Say an employee named Mary sets up report acceleration for a report, which leads to Splunk software building a summary for it. Then, a few days later, Joe designs a report that is nearly identical to Mary’s report, with a few variations. When Joe turns on report acceleration for the report and saves it, Splunk software automatically assigns it to the summary that was already built for Mary’s report, which means that Joe won't need to wait for the summary to be built.

• **Report acceleration features automatic backfill.** If for some reason you have a data interruption, Splunk software can detect this and automatically update or rebuild your summaries as appropriate.

• **Report acceleration summaries are stored alongside the buckets in your indexes.** Summary indexes, on the other hand, reside on the search head. Storing summaries in indexes at the bucket level enables Splunk Enterprise to easily handle the dilemma of late-arriving events--something that can force full rebuilds of summary indexes. Because summaries can simultaneously span both hot and warm buckets, they can summarize late-arriving data, because such data can only be added to hot buckets.

It's important to note that *not all searches qualify for report acceleration.* Only searches that utilize **transforming commands**--searches that transform their results into statistical tables and charts--are eligible. In addition, any commands used in the search before the transforming command must be **streaming commands.** This limitation is related to the fact that the summaries are built at the index level rather than the search head.

In Splunk Web, you can enable report acceleration for an eligible search when you save it as a report. You can enable report acceleration for an eligible existing report by:

- On the Reports page, expanding a row for a report and clicking **Edit** to open the **Edit Acceleration dialog.** If your report qualifies for acceleration and your permissions allow for report acceleration, the Edit Acceleration dialog will display a checkbox labeled **Accelerate Report.** Select it. The **Summary Range** field should appear. Select the range of time over which you plan to run the report, then click **Save.**
- in **Settings > Searches and reports** opening the detail page for a report, clicking **Accelerate this search** and setting a **Summary range.**

See Accelerate reports.
You use the Report acceleration summaries page in System to review and manage the summaries created through report acceleration.

See Manage report acceleration. This topic also explains how summaries work and includes examples of qualifying and non-qualifying searches.

**When should I use report acceleration?**

Report acceleration is good for just about any slow-completing report that has 100k or more hot bucket events and which meets the qualifying conditions outlined above.

**Data model acceleration**

You use data model acceleration to accelerate all of the fields defined in a data model. When a data model is accelerated, any pivot or report generated by that data model should complete much quicker than it would without the acceleration, even if the data model represents a significantly large dataset.

There are two types of data model acceleration, ad hoc and persistent. Ad hoc acceleration applies to a single dataset, is run over all time, and exists for the duration of a user’s pivot session, while persistent acceleration is turned on by an admin, happens in the background, and can be scoped to shorter time ranges such as a week or a month. Persistent acceleration is used any time a search is run against a dataset in an acceleration-enabled data model.

Data model acceleration makes use of Splunk’s high performance analytics store (HPAS) technology, which, in a manner similar to that of report acceleration, builds summaries alongside the buckets in your indexes. Also like report acceleration, persistent data model acceleration is easy to enable; you just click a checkbox for the data model you want to accelerate and select a summary range. Once you do this, Splunk software starts building a summary that spans the indicated range. When the summary is complete, any pivot, report, or dashboard panel that uses an accelerated data model dataset will run against the summary rather than the full array of _raw whenever possible, and result return time should be improved by a significant amount.

There are restrictions for persistent data model acceleration.

- **Persistent data model acceleration only applies to event dataset hierarchies and search dataset hierarchies based on root search datasets that only include streaming commands.** Dataset hierarchies based on root search datasets that nonstreaming commands and root
transaction datasets cannot be accelerated.
  • All data model datasets can benefit from "ad hoc" data model acceleration. See the subsection on this below.
  • **Once a data model is persistently accelerated it cannot be edited.** After you enable acceleration for a data model, the only way to edit it is to disable its acceleration.
  • **By default only users with admin permissions can persistently accelerate data models.**
  • **Data models that are private cannot be persistently accelerated.** You must share a data model with users of at least one app to make it eligible for acceleration.

In Splunk Web, you can enable data model acceleration for an eligible data model on the Data Models management page, which you can access in a variety of ways (including navigating to **Settings > Data Models**).

For more information about enabling persistent data model acceleration, see Manage data models.

For technical background information on data model acceleration and how the high performance analytics store works behind the scenes, see Accelerate data models.

**Ad hoc data model acceleration**

Ad hoc data model acceleration is a process that runs behind the scenes for all data model datasets that are not "persistently" accelerated beforehand. Unlike persistent data model acceleration, ad hoc data model acceleration applies to all dataset types, including root search datasets, root transaction datasets, and their children.

Whenever you build a pivot based on a dataset that isn't already accelerated, Splunk software will use ad hoc data model acceleration to build a temporary acceleration summary in a dispatch directory that exists only while you define the pivot in the Pivot Editor. The result is that as you fine-tune a particular pivot in the Pivot Editor you'll find that the pivot performance improves, returning results faster than it did when you first entered the editor.

This isn't as good as persistent data model acceleration, where summaries for the data model datasets are maintained on an ongoing basis, ensuring speedy performance from the moment you enter the Pivot Editor--but it's still helpful.

See Accelerate data models.
**When should I use persistent data model acceleration?**

If you are struggling with slow-completing pivots in the Pivot Editor and the source datasets for those pivots belongs to a topmost root event dataset hierarchy, you should consider enabling acceleration for that data model. It will ensure that the pivots based on those datasets return results faster than they would otherwise.

Furthermore, any report or dashboard panel that references a persistently accelerated data model dataset will also get this acceleration benefit (this will not happen with ad hoc data model acceleration).

**Report acceleration versus data model acceleration**

In general, data model acceleration is faster than report acceleration. However, there are specific kinds of searches that allow report acceleration to come out ahead of data model acceleration.

The more aggregating your transforming search is, the faster it can be. Report acceleration is especially fast when it runs with a search that aggregates down to one item per index bucket. For example, if you get a couple of billion events per day and you just want a monthly count and average, your return report acceleration will be better at this than data model acceleration. In this case you would be maxing out the aggregation capabilities of report acceleration.

Report acceleration and data model acceleration go about accelerating searches in similar ways. They both automatically preprocess events on the indexers, and they both create bucket-level acceleration summaries. But the general advantage for data model acceleration lies in how its summaries differ from those created by report acceleration.

Report acceleration is designed to create summaries that include precalculated statistics. Data model acceleration, on the other hand, builds its summaries in a format that is much more efficient to read, enabling Splunk software to calculate the statistics on demand without giving up performance. So if the search is a relatively complicated one you'll be better off going with data model acceleration.

**Summary indexing**

**Summary indexing** is a method you can use to speed up long-running reports that do not qualify for report acceleration, such as reports that use search commands that are not streamable before the transforming command. It's similar to report acceleration in that it involves populating a data summary with
the results of a search, but in this case the data summary is actually a special **summary index** that is built and stored on the search head. This summary index is populated by a scheduled report that is *based on* the report that you'd like to accelerate and which has **Enable** selected for summary indexing in **Settings > Searches and Reports**.

For example, if the report you want to accelerate uses a **transforming command**, you can populate its summary index with a report that swaps the transforming command with a similar "si-" prefix summary indexing transforming command: `sichart`, `sitimechart`, `sistats`, `sitop`, and `sirare`.

There are two topics on summary indexing setup.

- **Use summary indexing for increased reporting efficiency** shows you the easy way of setting up summary indexes, with scheduled searches that use *si-* commands.
- **Configure summary indexes** covers the tricky and difficult method of summary index setup with `addinfo`, `collect`, and `overlap` commands. You should only use this latter method if you're comfortable setting up searches that take aggregated statistics into account.

Summary indexing volume is not counted against your license, even if you have multiple summary indexes.

**When should I use summary indexing?**

If the report you're using qualifies for report acceleration, it's almost always preferable to use that method of speeding up the performance of large data volume searches.

You might want to use summary indexing instead of report acceleration if:

- The primary report you want to accelerate includes nonstreamable commands before a transforming command (just as with report acceleration, reports that populate summary indexes must involve transforming commands).
- You would like to run any report against a particular summary index, simply by including `index=<summary_index_name>` in your search string. (Under report acceleration, Splunk software automatically decides whether a report can run against a specific data summary.)
- Your raw data rolls more frequently than your reporting window (e.g. your retention policy is 6 months but you want to power a panel in a dashboard from data for the last year). Summary indexes generally take up less
space than the events they aggregate and can be retained separately and for greater durations.

**Batch mode search**

Batch mode search is a feature that improves the performance and reliability of transforming searches. For transforming searches that don't require the events to be time-ordered, running in batch mode means that the search executes bucket-by-bucket (in batches), rather than over time. In certain reporting cases, this means that the transforming search can complete faster. Additionally, batch mode search improves the reliability for long-running distributed searches, which can fail when an indexer goes down while the search is running. In this case, Splunk software attempts to reconnect to the missing peer and retry the search.

Transforming searches that meet the criteria for batch mode search include:

- Generating and transforming searches (stats, chart, etc.) that do not include the `localize` or `transaction` commands in the search.
- Searches that are not real-time and not summarizing searches.
- Non-distributed searches that are not stateful streaming. (A `streamstats` search is an example of a stateful streaming search.)

Batch mode search is invoked from the configuration file, in the `[search]` stanza of `limits.conf`. Use the search inspector to determine whether or not a transforming search is running in batch mode.

See Configure batch mode search.