Using Splunk: Performance Essentials

Splunk is powerful and versatile IT search software that takes the pain out of tracking and utilizing the information in your data center. If you have Splunk, you won't need complicated databases, connectors, custom parsers or controls--all that's required is a web browser and your imagination. Splunk handles the rest.

Use Splunk to:

- Continually index all of your IT data in real time.
- Automatically discover useful information embedded in your data, so you don't have to identify it yourself.
- Search your physical and virtual IT infrastructure for literally anything of interest and get results in seconds.
- Save searches and tag useful information, to make your system smarter.
- Set up alerts to automate the monitoring of your system for specific recurring events.
- Generate analytical reports with interactive charts, graphs, and tables and share them with others.
- Share saved searches and reports with fellow Splunk users, and distribute their results to team members and project stakeholders via email.
- Proactively review your IT systems to head off server downtimes and security incidents before they arise.
- Design specialized, information-rich views and dashboards that fit the wide-ranging needs of your enterprise

The main features of using Splunk can be summarized as follows:

- Index new data
- Search and investigate
- Capture knowledge
- Automate monitoring
- Analyze and report

Effects of bad search on Splunk

Here’s an example of a bad search to run in Splunk:

correlationId="sfuixamk:gzxtqula:00000000:00015047"

This is going to do a search for the provided correlationId in all logs that Splunk knows about. CorrelationId is not a special field, and this is going to be treated as a text search – very expensive. When you don’t preface a search to narrow down on your source, you’re forcing Splunk to have to search through ALL logs that it knows about. Splunk doesn’t “index” all of the contents of those files – only key fields. Most of the work is done at runtime on actual files, so a poorly formed search is going to waste a lot of resources unnecessarily.

Below we discuss some of the simple rules of thumb to help you write searches that will run more efficiently. Many factors can affect the speed of your searches: the volume of data that you are searching, how you've constructed your searches, whether or not you've planned your deployment sufficiently to handle the number of users running searches at the same time, and so on. The key to optimizing your search speed is to make sure that Splunk isn't doing more work than necessary.

1) Types of Searches:

The recommendations for optimizing searches vary depending on the type of search that you run and the characteristics of the data you're searching. In general, we describe searches based on what you are trying to do: retrieve events or generate reports. If the events you want to retrieve occur frequently in the dataset, we call it a *dense search*. If the events you want to retrieve are rare in the dataset, we call it a *sparse search*.

Raw event searches

Raw event searches return events from a Splunk index without any additional processing to the events that are retrieved. The best rule of thumb to follow when retrieving events from the index is to be specific about the events that you want to retrieve. You can do this with keywords and field/value pairs that are unique to the events. One thing to keep in mind is that sparse searches against large volumes of data will take longer than dense searches against the same data set.

- **Narrow down your search as much as possible from the start and limit the data that has to be pulled from disk to an absolute minimum.** For example, if you're only interested in Web access events, restrict your search to the specific host, index, or source type for that data.
- If you rarely search across more than one type of data at a time, **partition your different types of data into separate indexes and restrict your searches to the specific index**. For example, store Web access data in one index and firewall data in another.
- **Limit your search to the specific time window you need.** For example, to see what might have led to errors a few minutes ago, search within the last hour '-1hr', not the last week '-1w'.

Report-generating searches

Report-generating searches perform additional processing on events after they've been retrieved from an index. This processing can include filtering, transforming, and other operations using one or more statistical functions against the set of results. Because this processing occurs in memory, the more restrictive and specific you are when specifying the events to retrieve from disk, the faster the search will be.

- **If you are building a report, start your search from the Advanced Charting view instead of the timeline view.** The timeline view requires a lot of processing to calculate and build the timeline. When you run a search from the Advanced Charting view, it disables preview and the processing overhead associated with it.
- Reports rely on fields, so all the optimization rules for fields apply.

2) Use fields in your search

Searches with fields are faster when they use fields that have already been extracted (indexed fields) instead of fields extracted at search time.

- **Leverage indexed and default fields whenever you can to help search or filter your data efficiently.**

  At index time, Splunk extracts a set of default fields that are common to each event; these fields include host, source, and sourcetype. Use these fields to filter your data as early as possible in the search so that processing is done on a minimum amount of data. For example, if you're building a report on web access errors, search for those specific errors before the reporting command:
  Sourcetype = access_* (status=4 OR status=5*) | stats count by status*

- **Field extractions at search time add processing overhead.'**

  If you don't need additional fields in your search, turn off the *Discover Fields* option in the timeline view or use the fields command to specify only the fields that you want to see in your results.

3) Summarize your data

It can take a lot of time to search through very large data sets. If you regularly generate reports on large volumes of data, **use** summary indexing to pre-calculate the values that you use most often in your reports. Schedule saved searches to collect metrics on a regular basis, and report on the summarized data instead of on raw data.

4) Use the search job inspector

The Search Job Inspector is a tool you can use both to troubleshoot the performance of a search and to understand the execution costs of knowledge objects such as event types, tags, lookups, and other components within the search. It dissects the behavior of your searches so that you can better understand how to optimize them.

5) Best Practices

You can help Splunk get more out of your logs by following these best practices.

Use clear key-value pairs

One of the most powerful features of Splunk is its ability to extract fields from events when you search, creating structure out of unstructured data. To make sure field extraction works as intended, use the following string syntax (using spaces and commas is fine):

key1=value1, key2=value2, key3=value3 . . .

If your values contain spaces, wrap them in quotes (for example, username="bob smith").

This might be a bit more verbose than you are used to, but the automatic field extraction is worth the size difference.

Create human-readable events

- Avoid using complex encoding that would require lookups to make event information intelligible.
- Use human-readable timestamps for every event

  The correct time is critical to understanding the proper sequence of events. Timestamps are critical for debugging, analytics, and deriving transactions. Splunk will automatically timestamp events that don't include them using a number of techniques, but its best that you do it.

- Use the most verbose time granularity possible.

  Put the timestamp at the beginning of the line--the farther you place a timestamp from the beginning, the more difficult it is to tell it's a timestamp and no other data.

- Include a time zone, preferably a GMT/UTC offset.

  Time should be rendered in microseconds in each event. The event could become detached from its original source file at some point, so having the most accurate data about an event is ideal.

Use unique identifiers (IDs)

Unique identifiers such as transaction IDs and user IDs are tremendously helpful when debugging and even more helpful when you are gathering analytics. Unique IDs can point you to the exact transaction. Without them, you might only have a time range to use. When possible, carry these IDs through multiple touch points and avoid changing the format of these IDs between modules. That way, you can track transactions through the system and follow them across machines, networks, and services.

Log in text format

Avoid logging binary information because Splunk cannot be meaningfully search or analyze binary data. Binary logs might seem preferable because they are compressed, but this data requires decoding and won't segment. If you must log binary data, place textual meta-data in the event so that you can still search through it. For example, don't log the binary data of a JPG file, but do log its image size, creation tool, username, camera, GPS location, and so on.

Avoid using XML and JSON

Avoid formats with multi-depth nesting because they aren't human-readable and require more work to parse. Occasional XML is fine for dumping the value of something that exists in your code, but don't make a habit out of it.

Log more than just debugging events

Put semantic meaning in events to get more out of your data. Log audit trails, what users are doing, transactions, timing information, and so on. Log anything that can add value when aggregated, charted, or further analyzed. In other words, log anything that is interesting to the business.

Use categories

For example, use INFO, WARN, ERROR, DEBUG, and so on.

Keep multi-line events to a minimum

Multi-line events generate a lot of segments, which can affect indexing, and search speed, as well as disk compression. Consider breaking multi-line events into separate events.

Summary

When you're searching in Splunk with simple searches, you're searching ALL of the logs that Splunk knows about. You should always use these guidelines when searching in Splunk:

1. Preface your search with additional criteria to narrow down only search logs files relevant to you (source, sourcetype, logLevel)
2. Narrow your search to as small of a time frame as possible
3. Turn off "Field Discovery" unless it is necessary for your search.

Here's an example of a bad search to run in Splunk:

correlationId ="sfuixamk:gzxtqula:00000000:00015047"

That search can be improved by using one of the following variations to help narrow it down to only your application's log files:

Source ='yourApp' sourcetype=eventing logLevel=error correlationId="sfuixamk:gzxtqula:00000000:00015047"

This improved searches follow the guidance listed above, and will perform much better and user few resources.