# Table of Contents

**Introduction**
- Introduction to capacity planning for Splunk Enterprise ........................................... 1

**Hardware capacity planning**
- Components of a Splunk Enterprise deployment ................................................................. 2
- Dimensions of a Splunk Enterprise deployment ................................................................. 4
- How incoming data affects Splunk Enterprise performance .............................................. 6
- How indexed data affects Splunk Enterprise performance ................................................. 6
- How concurrent users affect Splunk Enterprise performance ........................................... 7
- How saved searches / reports affect Splunk Enterprise performance ............................... 7
- How search types affect Splunk Enterprise performance .................................................. 8
- How Splunk apps affect Splunk Enterprise performance ................................................... 9
- How Splunk Enterprise calculates disk storage ................................................................. 10
- Estimate your storage requirements ................................................................................. 11

**Scale your Splunk Enterprise Deployment**
- Distribute indexing and searching ..................................................................................... 13
- How concurrent users and searches impact performance ................................................. 15

**Performance Reference**
- Reference hardware ........................................................................................................... 19
- Determine when to scale your Splunk Enterprise deployment ........................................... 26
- Summary of performance recommendations ..................................................................... 28
- Forwarder-to-indexer ratios ............................................................................................... 29
- Parallelization settings ...................................................................................................... 30
Introduction

Introduction to capacity planning for Splunk Enterprise

You can expand Splunk Enterprise to meet almost any capacity requirement. To take advantage of this scaling capability requires planning. This manual discusses high-level hardware guidance for Splunk Enterprise deployments and describes how Splunk Enterprise uses hardware resources in different situations.

New for Splunk Enterprise version 6.2 and later, this manual supersedes guidance about capacity planning in the Installation and Distributed Deployment manuals. It provides information about reference hardware and a performance checklist to determine when and how you should scale your deployment based on your needs.

Before you decide on hardware for Splunk Enterprise, see the following information:

- Review Components of a Splunk Enterprise Deployment in this manual for a description of the elements in a Splunk Enterprise installation.

- Read about the dimensions of a Splunk Enterprise deployment, how those dimensions impact performance, and how to maximize performance.

- Learn about the basic building block of a Splunk Enterprise deployment in Reference hardware in this manual.
Hardware capacity planning

Components of a Splunk Enterprise deployment

The simplest deployment is the one you get by default when you first install Splunk Enterprise on a machine: a standalone instance that handles both indexing and searching. You log into Splunk Web or the CLI on the instance and configure data inputs to collect machine data. You then use the same instance to search, monitor, alert, and report on the incoming data.

You can also deploy specialized instances of Splunk Enterprise on multiple machines to address your load and availability requirements. These specialized instances are called "components". This section introduces the types of components. See the Distributed Deployment manual, particularly the topic, Scale your deployment with Splunk Enterprise components.

Indexer

Splunk indexers provide data processing and storage for local and remote data and host the primary Splunk data store. See How indexing works in the Managing Indexers and Clusters manual for more information.

Search head

A search head is a Splunk Enterprise instance that distributes searches to indexers (referred to as "search peers" in this context). Search heads can be either dedicated or not, depending on whether they also perform indexing. Dedicated search heads don't have any indexes of their own, other than the
usual internal indexes. Instead, they consolidate and display results that originate from remote search peers.

To configure a search head to search across a pool of indexers, see What is distributed search in the Distributed Search Manual

**Forwarder**

Forwarders are Splunk instances that forward data to remote indexers for data processing and storage. In most cases, they do not index data themselves. See the About forwarding and receiving topic in the Forwarding Data manual.

**Deployment server**

A Splunk Enterprise instance can also serve as a deployment server. The deployment server is a tool for distributing configurations, apps, and content updates to groups of Splunk Enterprise instances. You can use it to distribute updates to most types of Splunk components: forwarders, non-clustered indexers, and non-clustered search heads. See About deployment server and forwarder management in the Updating Splunk Enterprise Instances manual.

**Functions at a glance**

<table>
<thead>
<tr>
<th>Functions</th>
<th>Indexer</th>
<th>Search head</th>
<th>Forwarder</th>
<th>Deployment server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indexing</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Direct search</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forward to indexer</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Deploy configurations</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

**Index replication and indexer clusters**

An indexer cluster is a group of indexers configured to replicate each others’ data, so that the system keeps multiple copies of all data. This process is known as index replication. By maintaining multiple, identical copies of data, indexer clusters prevent data loss while promoting data availability for searching.

Splunk Enterprise clusters feature automatic failover from one indexer to the next. This means that, if one or more indexers fail, incoming data continues to
get indexed and indexed data continues to be searchable.

In addition to enhancing data availability, clusters have other features that you should consider when you are scaling a deployment, for example, a capability to coordinate configuration updates easily across all indexers in the cluster. Clusters also include a built-in distributed search capability. See About clusters and index replication in the Managing Indexers and Clusters of Indexers manual.

**Dimensions of a Splunk Enterprise deployment**

A Splunk Enterprise deployment has many dimensions. These scenarios determine whether a single reference machine can handle indexing and search load.

In some cases, a single reference machine can collect, store, and search data efficiently. In other cases, consider adding machines to your Splunk Enterprise deployment to increase performance. Below is a list of items that can have a significant impact on Splunk Enterprise performance.

- **Amount of incoming data.** The more data you send to Splunk Enterprise, the more time it needs to process the data into events that you can search, report, and generate alerts on.

- **Amount of indexed data.** As the amount of data stored in a Splunk Enterprise index increases, so does the I/O bandwidth needed to store data and provide results for searches.

- **Number of concurrent users.** If more than one person at a time uses an instance of Splunk Enterprise, that instance requires more resources for those users to perform searches and create reports and dashboards.

- **Number of saved searches.** If you plan to invoke a lot of saved searches, Splunk Enterprise needs capacity to perform those searches promptly and efficiently. A higher search count over a given period of time requires more resources.

- **Types of search you use.** Almost as important as the number of saved searches is the types of search that you run against a Splunk Enterprise instance. There are several types of search, each of which affects how the indexer responds to search requests.
• **Whether or not you run Splunk apps.** Splunk apps and solutions can have unique performance, deployment, and configuration considerations. If you plan to run apps, consider the resource requirements of the apps the you are using. See the documentation for the app for more information.

**How do these dimensions impact overall performance?**

While these factors have an impact on the basic sizing requirements of your Splunk Enterprise deployment, addressing each of them individually does not guarantee peak performance gain for the deployment. You must discover through trial how these factors correlate with one another in your specific application.

For example, if your Splunk Enterprise deployment calls for a low amount of indexing but has a high number of concurrent users, it has significantly different resource needs than a setup with a low number of concurrent users and a high amount of daily indexing volume. Additionally, as both user count and amount of indexed data rise, you must distribute the environment across multiple servers to maintain a similar performance level. Search types complicate matters, because some searches strain available CPU resources, while others depend on the speed of the disk subsystem.

**When should I scale my Splunk Enterprise deployment?**

You must understand how the deployment dimensions described in this topic apply to your specific use case. Answer the following questions, and then refer to the performance checklist in this manual to determine when you should add more hardware resources:

- How much data do you expect to index daily?
- How much data do you need to retain and for how long?
- How many users do you expect to search through the data at any one time?
- Do you plan to use certain specific searches more than once?
- Do you want or need to use a Splunk app to present or manipulate your data?

The key to a well-performing installation is to develop a plan early in the deployment cycle to account for both your initial outlay of hardware resources and the addition of resources when the deployment scales up.
How incoming data affects Splunk Enterprise performance

A reference Splunk Enterprise indexer can index a significant amount of data in a short period of time: over 20 megabytes of data per second or over 1.7 terabytes per day. This level of indexing occurs if the server is doing nothing else but consuming data.

Because Splunk Enterprise instances do more than index, consider this figure the maximum throughput for an indexer. Performance changes depending on the size and amount of incoming data. Larger events slow down indexing performance. As events increase in size, the indexer uses more system memory to process and index them.

If you need more indexing capacity than a single indexer can provide, add indexers into the deployment to account for the increased demand.

See the topics in this chapter to learn how other factors impact this performance figure.

How indexed data affects Splunk Enterprise performance

After Splunk Enterprise consumes data and places it into indexes, those indexes grow and take up disk space. As the indexes grow and available disk space decreases, Splunk Enterprise takes more time to index incoming data because the indexer's disk subsystem takes more time to find space to store the data.

This growth has an impact on search, as well. On a single indexer, disk throughput splits between indexing, which is ongoing, and search requests, which are interrupts based on requests scheduled by users. As indexes grow, search slows down because the disk subsystem needs to account for search requests, and it also needs to handle increasingly longer requests to store incoming data. Depending on the type of search, those kinds of requests can be I/O-intensive.
How concurrent users affect Splunk Enterprise performance

A reference indexer needs to dedicate one of its available CPU cores for every search that a user invokes for as long as the search is active. If multiple users are logged in and running searches, the number of available CPU cores can be exhausted quickly.

These figures assume that CPUs are idle when they receive a login or search request. This does not account for other system requests or CPU cores used by Splunk Enterprise to index data. If they are processing any other system requests, then the load splits across other available CPUs.

As CPU cores are used up, all activities on an indexer slow down as the computer splits processing time between indexing, search, and handling on-line users. Only additional indexers can increase capacity for all three functions of Splunk Enterprise operation.

How saved searches / reports affect Splunk Enterprise performance

On a reference indexer, a saved search or report consumes about 1 CPU core and a specified amount of memory while it executes. It behaves like an ad-hoc search. A saved search also increases the amount of disk I/O temporarily as the disk subsystem looks through the indexes to fetch data.

Each additional saved search that executes at the same time consumes an additional CPU core. This consumption is separate from CPU usage from the operating system and Splunk Enterprise indexing and storage processes.

If more saved searches execute than can be accepted for processing, they wait in a queue until they can be serviced. Splunk Enterprise also warns you when the system reaches the maximum number of queued saved searches. When searches queue up, search results return more slowly.

Adding search heads provides additional CPU cores to run more concurrent searches. Adding indexers helps scale with the increased search load and concurrency that comes from adding search heads. Adding RAM to your existing machines helps with concurrent searches but does not give you additional search capacity.
How search types affect Splunk Enterprise performance

You can invoke four types of searches against data stored in a Splunk Enterprise index. Each search type impacts the indexer in a different way.

The following table summarizes the different search types. For dense and sparse searches, Splunk Enterprise measures performance based on number of matching events. With super-sparse and rare searches, performance is measured based on total indexed volume.

<table>
<thead>
<tr>
<th>Search type</th>
<th>Description</th>
<th>Ref. indexer throughput</th>
<th>Performance impact</th>
</tr>
</thead>
</table>
| Dense       | Returns a large percentage (10% or more) of matching results for a given set of data in a given period of time. Dense searches usually tax a server's CPU first, because of the overhead required to decompress the raw data stored in a Splunk Enterprise index. Examples of dense searches include searches that use nothing but a wildcard character, or searching any index. Examples:

*  
index=m ?| stats count by fieldA  
index=a sourcetype=b ?| timechart count by myfield |
| Up to 50,000 matching events per second. | CPU-bound |
| Sparse      | Returns a smaller amount of results for a given set of data in a given period of time (anywhere from .01 to 1%) than do dense

<p>| Up to 5,000 matching events per second. | CPU-bound |</p>
<table>
<thead>
<tr>
<th>Search type</th>
<th>Description</th>
<th>Ref. indexer throughput</th>
<th>Performance impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Super-sparse</td>
<td>Returns a small number of results from each index <strong>bucket</strong> that matches the search. A super-sparse search is I/O intensive because the indexer must look through all of the buckets of an index to find the results. If you have a large amount of data stored on your indexer, there are a lot of buckets, and a super-sparse search can take a long time to finish.</td>
<td>Up to 2 seconds per index bucket.</td>
<td>I/O bound</td>
</tr>
<tr>
<td>Rare</td>
<td>Similar to a super-sparse search, but receives assistance from <strong>bloom filters</strong>, which help eliminate index buckets that do not match the search request. Rare searches return results anywhere from 20 to 100 times faster than does a super-sparse search.</td>
<td>From 10 to 50 index buckets per second.</td>
<td>I/O bound</td>
</tr>
</tbody>
</table>

**How Splunk apps affect Splunk Enterprise performance**

A single Splunk Enterprise indexer can run multiple apps simultaneously. Splunk Enterprise includes several apps which it runs at the same time.

However, the more complex apps offer advanced views that require the use of summarizing and accelerating searches that run in the background. The more background processing an app needs, the more likely you must distribute the processing load across multiple machines.

Many apps require a distributed Splunk Enterprise deployment by design. Whether it is a case of universal forwarders fetching data and sending it to a single central instance, or many indexers and search heads connected together
and serving up reports, dashboards, or alerts, Splunk apps often need more than one server to realize both maximum performance and potential in the enterprise.

**How Splunk apps affect resource requirements**

If you use a Splunk app or solution that gets knowledge by executing a large number of saved searches, then you can overwhelm a single-server Splunk Enterprise instance. Multiple searches quickly exhaust available CPU resources on an indexer. See *Accommodate many simultaneous searches* in this manual.

When you install an app or solution, read the system requirements outlined in that app or solution's documentation. If the information is not available, contact the authors of the app or solution to get information about what you need to run the app properly.

**How Splunk Enterprise calculates disk storage**

At a high level, Splunk calculates total disk storage as follows:

\[
( \text{Daily average indexing rate} ) \times ( \text{retention policy} ) \times \frac{1}{2}
\]

Splunk Enterprise stores raw data at up to approximately half its original size with compression. On a volume that contains 500GB of usable disk space, you can store nearly six months' worth of data at an indexing rate of 5GB/day or ten days' worth at a rate of 100GB/day.

If you need additional storage, you can opt for either more local disks, which is required for frequent searching, or you can use attached or network storage, which is acceptable for occasional searching. Low-latency connections over NFS or SMB/CIFS (Server Message Block/Common Internet File System) are acceptable for searches over long time periods where instant search returns can be compromised to lower cost per GB.

**Important:** Shares mounted over a Wide Area Network (WAN) connection or on standby storage such as tape are never suitable storage choices for Splunk Enterprise operations.
Estimate your storage requirements

This topic describes how to estimate the size of your Splunk Enterprise index, so that you can plan your storage capacity requirements.

When Splunk Enterprise indexes your data, it creates two main types of files: the "rawdata" file that contains the original data in compressed form and the index files that point to this data. (It also creates a few metadata files, which don't consume much space.) With a little experimentation, you can estimate how much index disk space you will need for a given amount of incoming data.

Typically, the compressed rawdata file is 10% the size of the incoming, pre-indexed raw data. The associated index files range in size from approximately 10% to 110% of the rawdata file. The number of unique terms in the data affect this value.

Depending on the data's characteristics, you might want to tune your segmentation settings, as described in About segmentation in the Getting Data In Manual.

The best way to get an idea of your space needs is to experiment by indexing a representative sample of your data, and then checking the sizes of the resulting directories in $SPLUNK_HOME/var/lib/splunk/defaultdb.

On *nix systems, follow these steps

Once you've indexed your data sample:

1. Go to $SPLUNK_HOME/var/lib/splunk/defaultdb/db.

2. Run du -ch hot_v* and look at the last total line to see the size of the index.

On Windows systems, follow these steps

1. Download the du utility from Microsoft TechNet.

2. Extract du.exe from the downloaded ZIP file and place it into your %SYSTEMROOT% or %WINDIR% folder.

Note: You can also place it anywhere in your %PATH%.

3. Open a command prompt.
4. Once there, go to `%SPLUNK_HOME%\var\lib\splunk\defaultdb\db`.

5. Run `del %TEMP%\du.txt & for /d %i in (hot_v*) do du -q -u %i\rawdata | findstr /b "Size:" >> %TEMP%\du.txt`.

6. Open the `%TEMP%\du.txt` file. You will see Size: n, which is the size of each rawdata directory found.

7. Add these numbers together to find out how large the compressed persisted raw data is.

8. Next, run `for /d %i in (hot_v*) do dir /s %i`, the summary of which is the size of the index.

9. Add this number to the total persistent raw data number.

This is the total size of the index and associated data for the sample you have indexed. You can now use this to extrapolate the size requirements of your Splunk Enterprise index and rawdata directories over time.

**Answers**

Have questions? Visit Splunk Answers to see what questions and answers other Splunk users had about data sizing.
Scale your Splunk Enterprise Deployment

Distribute indexing and searching

This topic discusses the reasons to distribute the components of your Splunk platform deployment.

Concepts of distributed indexing and searching

Designing a scalable architecture for the Splunk platform requires knowledge of the Splunk instance roles, and how they were intended to scale.

The two most common roles are the search head and the indexer. They represent the roles that carry the burden of managing user objects, searching, parsing, and data storage.

A Search head is responsible for:

- Hosting users.
- Storing user created objects.
- Scheduling searches and alerting.
- Providing visual feedback through dashboards and views.
- Enforcing access controls.

An Indexer is responsible for:

- Accepting data streams from forwarders.
- Parsing the data.
- Writing the data into buckets.
- Maintaining the buckets.
- Accepting search requests from the search heads.
- Searching the buckets and streaming results back to the search head.

A search head's tasks are primarily CPU bound. As more users and more apps are added to a search head, the concurrent search load climbs quickly and hits a limit. The limit represents the aggregate search load across all users and apps to a search head's CPU cores.

Adding search heads to the deployment increases the aggregate CPU resources, increasing the aggregate search concurrency and the number of active users and apps supported in the environment.
An indexer's tasks are primarily I/O bound. As more forwarders are added to the network, there are more concurrent data streams to accept, and more data to parse before writing. In addition, the search requests require I/O access and processor time to analyze, collect, and return the requested data. As the data streams increase in volume and the concurrent search requests climb higher, the indexer hits a limit. The limit represents the aggregate search load from all search heads and indexing load from forwarders to an indexer's I/O capacity.

Adding indexers to the deployment increases the total aggregate I/O capacity and storage available to save data, reduces the data volume per indexer load, and reduces the impact of the search load by spreading it across more indexers.

**Scaling the Splunk platform**

A typical Splunk platform deployment plan is based upon 2 points: indexed data volume per day, and estimated search load. User counts are often used as a proxy for search load. For example, one active user with admin-level search concurrency can sustain the same load on the Splunk platform deployment as several users at a lower role level.

Most Splunk implementations are built around a handful of users and a few apps searching hundreds of gigabytes of data. In that scenario, adding indexers is the preferred method of scaling. The same rule applies when implementing a indexer cluster.

As the search head gains more users, the CPU limitations will become apparent as searches may skip, and users experience slower search result speed. Adding another search head to your distributed deployment does not guarantee improved search performance. As the user count increases, indexers must be added to maintain search performance. For a table with scaling guidelines, see Summary of performance recommendations in this manual.

When planning for a user count at 50 or more, consider implementing a search head cluster to absorb a high level of users while adding redundancy to the search tier.

**Scaling performance**

As your indexers consume data, they store it in buckets, which are the individual elements of an index. As more data comes in, the number of buckets increases. As the number of buckets increases, the indexer must manage the buckets by "rolling" them to make room for new incoming data. This procedure takes up I/O cycles, which reduces the resources available to fetch events for search
requests. The impact is noticeable for index buckets that hold smaller amounts of data.

Avoid configuring many indexes comprised of small buckets. For examples utilizing the maxDataSize bucket setting, see indexes.conf.example in the Splunk Enterprise Admin manual.

The number and types of search also impact indexer performance. Most search types leverage an indexer's disk subsystem, but a few will use more CPU. For information on simultaneous searches, see Accommodate concurrent users and searches in this manual.

If the hardware allocated for indexers exceeds the reference machine specifications, consider reviewing and implementing one of the Parallelization settings to improve the performance for specific use cases.

Use the monitoring console to monitor and track resource usage across the Splunk platform environment. For more details, see About the monitoring console in Monitoring Splunk Enterprise.

How concurrent users and searches impact performance

The largest performance factor in a Splunk Enterprise deployment are:

- The number of concurrent users.
- The number of concurrent searches.
- The types of searches used.

A user that submits a search request will use one CPU core on each indexer until the search is complete. Any additional searches that the user submits also account for one CPU core. You can adjust the number of global concurrent searches that a machine can run. See Expected performance and known limitations of real-time searches and reports in the Splunk Enterprise Search manual.

The type of search a user invokes also impacts hardware resource usage. See How search types affect Splunk Enterprise performance.
How to maximize search performance

To accommodate the resource overhead of running many concurrent searches, add additional indexers, and maximize the physical memory available to the indexers. The indexers do the bulk of the work in search operations, such as identifying the data requested, reading the data from disk, decompressing it, filtering the data, and streaming the results.

For example, if a search uses 200MB of memory, and there are 48 concurrent search requests, that equates to about 10GB of memory to meet the search load not including the OS requirements. The amount of available memory is an important resource to monitor. While performance on an indexer declines gradually with increased CPU usage from concurrent search jobs, it drops dramatically when all available physical memory is exhausted.

Search performance: A basic scenario

The aggregate run time of all searches increases as the number of available CPU cores on the indexer decreases. For example, on an indexer with no load and 12 available cores, the first searches to arrive for processing can complete within a short period of time. For this scenario, all searches will run to completion within 10 seconds.

12 concurrent searches: One indexer with 12 cores and no data being indexed.

<table>
<thead>
<tr>
<th>No. of concurrent searches</th>
<th>/ No. of avail. cores</th>
<th>= No. of searches per core</th>
<th>No. of sec. per individual search</th>
<th>= Approx. time (sec.) to complete all searches</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>12</td>
<td>1</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

When there are 48 searches to run concurrently, the total time to complete all searches increases significantly.

48 concurrent searches: One indexer with 12 cores and no data being indexed.

<table>
<thead>
<tr>
<th>No. of concurrent searches</th>
<th>/ No. of avail. cores</th>
<th>= No. of searches per core</th>
<th>No. of sec. per individual search</th>
<th>= Approx. time (sec.) to complete all searches</th>
</tr>
</thead>
</table>
Because the indexers do the bulk of the work in search operations, such as identifying the data requested, reading the data from disk, decompressing it, filtering some data, and streaming the results, it is best practice to add indexers to decrease the total amount of time to return all search results.

A deployment with more indexers provides an aggregate increase in cores to improve the search completion time when there are many concurrent searches. When there are more cores available than concurrent searches, the cores can be used by Splunk Enterprise to perform maintenance operations, or can remain idle.

12 concurrent searches: 4 indexers with 12 cores per indexer and no data being indexed.

<table>
<thead>
<tr>
<th>No. of concurrent searches</th>
<th>/ No. of avail. cores</th>
<th>= No. of searches per core</th>
<th>No. of sec. per individual search</th>
<th>= Approx. time (sec.) to complete all searches</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>48</td>
<td></td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

1. By default, a search cannot take advantage of multiple cores.

48 concurrent searches: 4 indexers with 12 cores per indexer and no data being indexed.

<table>
<thead>
<tr>
<th>No. of concurrent searches</th>
<th>/ No. of avail. cores</th>
<th>= No. of searches per core</th>
<th>No. of sec. per individual search</th>
<th>= Approx. time (sec.) to complete all searches</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>48</td>
<td>1</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Search performance: Indexing data scenario

In an active deployment, the system is not sitting idle while searches arrive. If an indexer ingests 150GB/day of data, then it will use up to 4 of the available cores for indexing processes. With fewer cores available, the time it takes to return all search results increases.
12 concurrent searches: One 12-core indexer, with 8 available cores.

<table>
<thead>
<tr>
<th>No. of concurrent searches</th>
<th>/ No. of avail. cores</th>
<th>= No. of searches per core</th>
<th>No. of sec. per individual search</th>
<th>= Approx. time (sec.) to complete all searches</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>8</td>
<td>2, as each core will remain in use until the prior search completes.</td>
<td>10</td>
<td>20</td>
</tr>
</tbody>
</table>

48 concurrent searches: One 12-core indexer, with 8 available cores.

<table>
<thead>
<tr>
<th>No. of concurrent searches</th>
<th>/ No. of avail. cores</th>
<th>= No. of searches per core</th>
<th>No. of sec. per individual search</th>
<th>= Approx. time (sec.) to complete all searches</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>8</td>
<td>6</td>
<td>10</td>
<td>60</td>
</tr>
</tbody>
</table>

48 concurrent searches: 4 12-core indexers with 8 available cores.

<table>
<thead>
<tr>
<th>No. of concurrent searches</th>
<th>/ No. of avail. cores</th>
<th>= No. of searches per core</th>
<th>No. of sec. per individual search</th>
<th>= Approx. time (sec.) to complete all searches</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>32</td>
<td>2, as each core will remain in use until the prior search completes.</td>
<td>10</td>
<td>20</td>
</tr>
</tbody>
</table>

Having fewer indexers with greater core counts per indexer can decrease the amount of total time for search, but also reduces scaling efficiency by providing less aggregate IOPS for search operations.

Adding indexers reduces the indexing load on any one machine. Additionally, you reduce the search time, lower the impact of high search concurrency, and lower the impact of resource contention for I/O and memory.
Performance Reference

Reference hardware

The reference hardware specification is a baseline for scoping and scaling the Splunk platform for your use. It is a performance guide for handling search and indexing loads.

Reference host specification for single-instance deployments

The following requirements represent the basic building block of a Splunk Enterprise deployment.

- Intel x86 64-bit chip architecture
- 12 CPU cores at 2Ghz or greater speed per core
- 12GB RAM
- Standard 1Gb Ethernet NIC, optional second NIC for a management network
- Standard 64-bit Linux or Windows distribution

The requirements are the same for a Splunk Enterprise instance that forwards data or one that does not. There are two additional reference specifications that provide higher performance and search concurrency. These specs are described later in this topic.

Disk subsystem

The disk subsystem for a reference machine should be capable of handling a high number of average Input/Output Operations Per Second (IOPS).

IOPS are a measurement of how much data throughput a hard drive can produce. Because a hard drive reads and writes at different speeds, there are IOPS numbers for disk reads and writes. The average IOPS is the blend between those two figures.

The more average IOPS that a hard drive can produce, the more data it can index and search in a given period of time. While many variable items factor into the amount of IOPS that a hard drive can produce, the following are the most important:

- Its rotational speed in revolutions per minute.
• Its average latency, which is the amount of time it takes to spin its platters half a rotation.
• Its average seek time, which is the amount of time it takes to retrieve a requested block of data.

The drives that are capable of producing the highest IOPS have high rotational speeds and low average latency and seek times. Every drive manufacturer provides this information, and some provide much more.

For information on IOPS and how to calculate them, see Getting the hang of IOPS on the Symantec Connect Community.

This specification uses eight 146GB, 15,000 RPM, serial-attached SCSI (SAS) HDs in a Redundant Array of Independent Disks (RAID) 1+0 fault tolerance scheme as the disk subsystem. Each hard drive is capable of about 200 average IOPS. The combined array produces a little over 800 average IOPS.

Insufficient disk I/O is the most common limitation in Splunk infrastructure. For the best results indexing your data, review the disk subsystem requirements before provisioning your hardware.

**Maximum performance capability**

The maximum performance capabilities measure indexing and search performance independently, and do not represent the combined load of a typical Splunk use case. To review performance recommendations for a reference machine with indexing with search load, see Summary of performance recommendations.

**Indexing performance**

• Up to 20MB per second (1700GB per day) of raw indexing performance if no searching or other index-related activity occurs.

**Search performance**

• Up to 50,000 events per second for dense searches.
• Up to 5,000 events per second for sparse searches.
• Up to 2 seconds per index bucket for super-sparse searches.
• From 10 to 50 buckets per second for rare searches with bloom filters.

To find out more about the types of searches and how they affect Splunk Enterprise performance, see How search types affect Splunk Enterprise
Reference host specification for distributed deployments

As the number of active users increases along with the data ingestion rate, the architecture requirements change from a single instance to a distributed Splunk Enterprise environment. The search head and indexer roles have unique hardware recommendations.

**Dedicated search head**

A search head uses CPU resources more consistently than an indexer, but does not require the fast disk throughput or a large pool of local storage for indexing.

- Intel 64-bit chip architecture
- 16 CPU cores at 2Ghz or greater speed per core.
- 12GB RAM
- 2 x 300GB, 10,000 RPM SAS hard disks, configured in RAID 1
- A 1Gb Ethernet NIC, optional 2nd NIC for a management network
- A 64-bit Linux or Windows distribution

A search request uses up to 1 CPU core while the search is active. You must account for scheduled searches when you provision a search head in addition to ad-hoc searches that users run. More active users and higher concurrent search loads require additional CPU cores.

For a review on how searches are prioritized, see the topic Configure the priority of scheduled reports in the Reporting Manual. For information on scaling search performance, see How to maximize search performance.

**Indexer**

When you distribute the indexing process, the Splunk platform can scale to consume terabytes of data in a day. When you add more indexers, you distribute the work of search requests and data indexing across those indexers. This increases performance significantly.

As a reminder, here is the reference indexer specification:

**Reference host specification**

- Intel 64-bit chip architecture.
- 12 CPU cores at 2GHz or greater per core.
• 12GB RAM.
• Disk subsystem capable of a minimum of 800 average IOPS. For details, see the topic Disk subsystem.
• A 1Gb Ethernet NIC, with optional second NIC for a management network.
• A 64-bit Linux or Windows distribution.

Splunk has introduced two new specifications that help improve user experience by providing additional CPU cores for better indexing performance and search concurrency.

A single indexer carries the same disk I/O bandwidth requirements as a group of indexers.

**Mid-range specification**

The mid-range specification is similar to the base reference specification. This specification improves indexing capacity and search concurrency over a distributed Splunk Enterprise deployment.

• Intel 64-bit chip architecture
• 24 CPU cores at 2GHz or greater speed per core
• 64GB RAM
• Disk subsystem capable of a minimum of 800 average IOPS
• A 1Gb Ethernet NIC, with optional second NIC for a management network
• A 64-bit Linux or Windows distribution

**High-performance specification**

The high-performance specification is a further improvement upon the mid-range specification.

• Intel 64-bit chip architecture
• 48 CPU cores at 2GHz or greater speed per core
• 128GB RAM
• Disk subsystem capable of a minimum of 1200 average IOPS
• A solid state disk (SSD) subsystem as a minimum requirement for hot and warm index buckets
• A 1Gb Ethernet NIC with optional second NIC
• A 64-bit Linux or Windows distribution
Disk subsystem information for higher-performance specifications

When indexers retrieve data for searches, they do many disk seeks and bulk reads. At higher daily volumes, local disk might not provide cost-effective storage for the time frames where you want a fast search. Fast attached storage or networked storage, such as storage area networks (SAN) over fiber, can provide the required IOPS for each indexer in these cases.

When you plan your storage infrastructure, understand these key points:

- More disks (specifically, more spindles) are better for indexing performance.
- Total throughput of the entire system is important.
- The ratio of disks to disk controllers in a particular system should be higher, similar to how you provision a database host.

Ratio of indexers to search heads

There is no practical limitation on the number of search heads that an indexer can support, or on the number of indexers that a search head can search against. The use case determines what Splunk instance role (search head or indexer) the infrastructure needs to scale while maintaining performance. For a table with scaling guidelines, see Summary of performance recommendations.

Network latency limits for clustered deployments

A Splunk environment with search head or indexer clusters must have fast, low-latency network connectivity between clusters and cluster nodes. This is particularly important in environments that have clusters in multiple sites.

For indexer cluster nodes, network latency should not exceed 100 milliseconds. Higher latencies can significantly slow indexing performance and hinder recovery from cluster node failures.

Impact of network latency on clustered deployment operations.

<table>
<thead>
<tr>
<th>Network latency</th>
<th>Cluster Index time. 1 TB of data</th>
<th>Cluster node recovery time</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 100 ms</td>
<td>6202 s</td>
<td>143 s</td>
</tr>
<tr>
<td>300 ms</td>
<td>6255 s (+ 1%)</td>
<td>1265 s (+ 884%)</td>
</tr>
<tr>
<td>600 ms</td>
<td>7531 s (+ 21%)</td>
<td>3048 s (+ 2131%)</td>
</tr>
</tbody>
</table>
Network latency | Cluster Index time. 1 TB of data | Cluster node recovery time
---|---|---
The impact of latency can vary based on individual configurations.

For search head clusters, latency should not exceed 200 milliseconds. Higher latencies can impact how fast a search head cluster elects a cluster captain.

Confirm with your network administrator that the networks that will support the clustered Splunk environments meet or exceed these latency guidelines. If reduction of latency below these levels is not possible, then contact Splunk Support or Professional Services to discuss tuning cluster timeout settings on the cluster nodes to handle the increased latency.

**Premium Splunk app requirements**

Premium Splunk apps can demand greater hardware resources than the reference specifications in this topic provide. A Splunk Enterprise instance that runs a premium app such as Enterprise Security or Splunk IT Service Intelligence must have a disk subsystem that can produce a minimum of 1200 average IOPS.

Before architecting a deployment for a premium app, review the app documentation for scaling and hardware recommendations. The following list shows examples of some premium Splunk apps and their recommended hardware specifications.

- Splunk Enterprise Security
- Splunk IT Service Intelligence
- Splunk App for PCI

**Virtual hardware**

Splunk supports use of its software in virtual hosting environments. An indexer on a hypervisor (such as VMware) with reserved resources that meet one of the hardware specifications can consume data about 10 to 15 percent more slowly than an indexer hosted on a bare-metal host. Search performance in a virtual hosting environment is a close match to bare-metal computers.

The performance that a virtual host provides is a best-case scenario that does not account for resource contention with other active virtual hosts that share the same physical host or storage array. It also does not account for certain vendor-specific I/O enhancement techniques, such as Direct I/O or Raw Device Mapping.
For information on how to run Splunk Enterprise in a VMware virtual machine, see Deploying Splunk Enterprise Inside Virtual Environments on splunk.com.

**Splunk Enterprise, self-managed in the cloud**

Running Splunk Enterprise in the cloud is an alternative to running it on-premises using bare-metal hardware. Splunk Enterprise delivers similar performance on a cloud-based infrastructure as it does on bare-metal hardware. Depending upon the vendor and technologies used to provision cloud instances, there can be less resources available than the OS reports.

If you run Splunk Enterprise on an Amazon Web Services (AWS) instance:

- AWS measures CPU power on Elastic Compute Cloud (EC2) instances in virtual CPUs (vCPUs), not real CPUs.
- Each vCPU is a hyper thread of an Intel Xeon core on most AWS instance types. See AWS | Amazon EC2 | Instance Types on the AWS web site.
- As a hyper thread of a core, a vCPU acts as a core, but the physical core must schedule its workload among other workloads of other vCPUs that the physical core handles.

For indexing and data storage, note the following:

- If you choose to use Elastic Block Storage (EBS), the type of EBS volume you choose determines the amount of performance you get.
- Not all EBS volume types have the necessary IOPS to handle Splunk Enterprise operations.
- The "Provisioned IOPS" and "Magnetic" EBS volume types offer the best opportunity to get the IOPS that you need for indexing and searching. See EBS - Product Details on the AWS web site.
- Not every EC2 instance type offers the network throughput to the EBS volume that you need. To ensure that bandwidth you must either launch the instance as "EBS-optimized" or choose an instance type that provides a minimum of 10Gb of bandwidth. See Amazon EC2 Instance Configuration on the AWS web site.

For forwarding, note that the proximity of your cloud infrastructure to your forwarders can have a major impact on performance of the whole environment.

For recommendations on running Splunk Enterprise in AWS, see Deploying Splunk Enterprise On Amazon Webservices on splunk.com.
**Splunk Cloud**

Splunk offers its machine data platform and licensed software as a subscription service called Splunk Cloud. When you subscribe to the service, you purchase a capacity to index, store, and search your machine data. Splunk Cloud abstracts the infrastructure specification from you and delivers high performance on the capacity you have purchased.

To learn more about Splunk Cloud, visit the Splunk Cloud website.

**Considerations for deploying Splunk software on partner infrastructure**

Many hardware vendors and cloud providers have worked to create reference architectures and solution guides that describe how to deploy Splunk Enterprise and other Splunk software on their infrastructure. For your convenience, Splunk maintains a separate page where Splunk Technology Alliance Partners (TAP) may submit reference architectures and solution guides that meet or exceed the specifications of the documented reference hardware standard. See the Splunk Partner Solutions page on the Splunk website.

While Splunk works with TAPs to ensure that their solutions meet the standard, it does not endorse any particular hardware vendor or technology.

**Determine when to scale your Splunk Enterprise deployment**

Before you consider when and how to scale your environment, estimate how much data you need to index, and how many users are searching that data.

**Performance questionnaire**

This questionnaire begins with a single-instance Splunk Enterprise deployment based on the reference architecture described in the Reference machine for single-instance deployments topic. These guidelines help you decide when to distribute your Splunk platform deployment.
**Question 1:** Do you need to index more than 2GB of data per day?

**Question 2:** Do you need more than two users signed in at one time?

If you answer **No** to questions 1 and 2, then your Splunk platform instance can share a reference machine for distributed deployments with other Splunk platform services.

If you answer **Yes** to question 1 or 2, then proceed to Question 3.

**Note** When deploying Splunk Enterprise on Windows OS, do not utilize a host that provides Active Directory or Exchange services, or runs machine virtualization software. Those services are I/O intensive and can reduce Splunk Enterprise indexing and search performance.

**Question 3:** Do you need to index more than 300GB per day?

**Question 4:** Do you need more than four concurrent users?

If you answer **No** to questions 3 and 4, then a single dedicated Splunk Enterprise instance running on a reference machine can provide sufficient resources for the indexing and search workload. Go to Question 5.

If you answer **Yes** to question 3 or 4, then scale your Splunk Enterprise deployment to multiple machines to handle the increased demand of indexing and searching. Go to Question 5.

**Question 5:** Do you need more than 600GB of total storage?

See How Splunk Enterprise calculates disk storage.

If you answer **No**, then a single dedicated reference machine should be able to handle indexing and search workload, but you can consider adding additional storage to the machine to account for increased disk usage due to higher retention. Go to Question 6.

If you answer **Yes**, then scale your Splunk Enterprise deployment to multiple machines to handle the increased demand of indexing and searching. Go to Question 6.
**Question 6:** Do you want to create or run a Splunk app, alert, or solution that executes more than 8 concurrent saved searches?

**Question 7:** Do you need to search large quantities of data for a small set (less than 1 per cent) of results?

If you answer **No** to questions 6 and 7, you might not require multiple indexers in your Splunk Enterprise deployment at this time.

If you answer **Yes** to questions 6 or 7, then scale your Splunk Enterprise deployment to multiple machines to handle the increased demand of indexing and searching.

### Summary of performance recommendations

The Daily Indexing Volume table summarizes the performance recommendations that were given in the performance checklist. The table shows the number of reference machines that you need to index and search data in Splunk Enterprise, depending on the number of concurrent users and the amounts of data that the instance indexes.

An indexer that meets the reference hardware requirements can ingest up to 300GB/day while supporting a search load. For a review of the current reference hardware specifications, see Reference hardware in this manual.

The table is only a guideline. Modify these figures based on your use case. If you need help defining and scaling a Splunk platform environment, contact your Splunk Sales representative or Professional Services.

<table>
<thead>
<tr>
<th>Daily Indexing Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 2GB/day</td>
</tr>
<tr>
<td>Total Users: less than 4</td>
</tr>
<tr>
<td>1 combined instance</td>
</tr>
</tbody>
</table>
### Daily Indexing Volume

<table>
<thead>
<tr>
<th>Total Users: up to 8</th>
<th>1 combined instance</th>
<th>1 Search Head, 1 Indexers</th>
<th>1 Search Head, 2 Indexers</th>
<th>1 Search Head, 3 Indexers</th>
<th>1 Search Head, 8 Indexers</th>
<th>1 Search Head, 12 Indexers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Users: up to 16</td>
<td></td>
<td>1 Search Head, 1 Indexers</td>
<td>1 Search Head, 3 Indexers</td>
<td>2 Search Heads, 4 Indexers</td>
<td>2 Search Heads, 10 Indexers</td>
<td>2 Search Heads, 15 Indexers</td>
</tr>
<tr>
<td>Total Users: up to 24</td>
<td></td>
<td>1 Search Head, 2 Indexers</td>
<td>2 Search Heads, 3 Indexers</td>
<td>2 Search Heads, 6 Indexers</td>
<td>2 Search Heads, 12 Indexers</td>
<td>3 Search Heads, 18 Indexers</td>
</tr>
<tr>
<td>Total Users: up to 48</td>
<td></td>
<td>1 Search Head, 2 Indexers</td>
<td>2 Search Heads, 4 Indexers</td>
<td>2 Search Heads, 7 Indexers</td>
<td>3 Search Heads, 14 Indexers</td>
<td>3 Search Heads, 21 Indexers</td>
</tr>
</tbody>
</table>

### Forwarder-to-indexer ratios

Splunk Enterprise indexers are responsible for accepting data streams from internal and external sources, such as forwarders, and indexing that stream locally. Indexing the data requires lots of disk I/O bandwidth and some computing resources. Indexing capacity remains the top concern when you consider how many forwarders an indexer can handle.

The number of forwarders from which an indexer can accept data depends on several factors:

- Number of CPU cores on the machine. The number of cores should meet or exceed the reference standard.
- Number of disk spindles on the machine. The number of spindles should meet or exceed the reference standard.
- Whether the indexer runs Windows or *nix.
- The amount of data to be forwarded to the indexers.
- Whether the indexer also acts as a deployment server.
Forwarder-to-indexer ratios for a *nix indexer

Splunk Enterprise used the following setup to provide guidance for the number of forwarders that can connect to a *nix indexer:

- An indexer with 8 cores and 7GB of RAM and 4 x 420GB disks in RAID 0, running a 64-bit Linux OS.
- A high-speed local area network (LAN) operating at 100Mb/s or faster.
- All universal forwarders sent data that was not processed beforehand.

In these circumstances, an indexer was able to handle a minimum of 2000 forwarders and regularly handled as many as 5000 forwarders.

Performance was best when the server was configured to accept a high number of Unix file descriptors, typically three to four times the number of forwarders that the indexer could accept.

*Note:* These numbers are for guidance only. Results vary depending on the configuration of the indexers, forwarders, and network.

Parallelization settings

New settings are available in Splunk Enterprise to improve search and indexing performance.

Who can use these settings

The parallelization settings are designed to improve the performance of specific components in Splunk Enterprise. The parallelization features are intended for customers with excess CPU cores and I/O capacity to leverage their hardware for improved performance across the indexing tier. You can use these settings to allocate CPU resources to the most common uses for your Splunk platform environment, tuning the indexers to meet that demand.

Summary of settings

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch mode search parallelization</td>
<td>Allows a batch mode search to open additional search pipelines on each indexer, processing</td>
</tr>
</tbody>
</table>
If the indexers in your Splunk platform environment exceed the reference hardware specifications, you may review the use case and increase one parallelization settings up to the maximum recommended value. If your indexers are at or near capacity, changing the parallelization settings can have a negative impact on search and indexing performance. All parallelization settings require a service restart to take effect.

### Batch mode search parallelization

Batch mode searches are designed to search and return event data by bucket, instead of by time. By adding more batch search pipelines, multiple buckets are processed simultaneously, speeding the return of search results. Customers leveraging batch mode search parallelization can see a doubling of speed in returning batch mode search results.

<table>
<thead>
<tr>
<th>Setting name</th>
<th>Default</th>
<th>Maximum recommended value</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>batch_search_max_pipeline</td>
<td>1</td>
<td>2</td>
<td>Multiplies the number of search pipelines per batch mode</td>
</tr>
</tbody>
</table>
Adjusting the `batch_search_max_pipeline` setting in `limits.conf` to 2 multiplies the IO, processing, and memory used by batch mode searches on every indexer. A value of 2 provides the best performance increase, with higher values succumbing to diminishing returns. For configuration details, see Configure batch mode search parallelization in the Splunk Enterprise Knowledge Manager Manual.

Splunk administrators can use the monitoring console to monitor and track indexer resource use. For more details, see About the monitoring console in Monitoring Splunk Enterprise.

### Parallel summarization

There are two types of accelerated searches: data model accelerations and report accelerations. Both acceleration types create search results on disk beside each index bucket. When a scheduled acceleration search is unable to keep up with the data volume in an index, latency is introduced into the search results. By allowing the scheduler to run concurrent acceleration searches on the indexers, multiple buckets are processed simultaneously, speeding the creation of accelerated search results. Customers leveraging parallel summarization can see a doubling of speed in building accelerated search results.

#### Data model accelerations

<table>
<thead>
<tr>
<th>Setting name</th>
<th>Default</th>
<th>Maximum recommended value</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>acceleration.max_concurrent</code></td>
<td>3</td>
<td>3</td>
<td>Multiplies the number of scheduled acceleration searches per data model, per indexer.</td>
</tr>
</tbody>
</table>
The `acceleration.max_concurrent` setting in `datamodels.conf` defaults to 3, multiplying the IO, processing, and memory used while running scheduled acceleration searches on every indexer. A value of 3 provides the best performance increase, with higher values succumbing to diminishing returns. For configuration details, see Parallel Summarization in the Splunk Enterprise Knowledge Manager Manual.

**Report accelerations**

<table>
<thead>
<tr>
<th>Setting name</th>
<th>Default</th>
<th>Maximum recommended value</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>auto_summarize.max_concurrent</code></td>
<td>1</td>
<td>2</td>
<td>Multiplies the number of scheduled acceleration searches per search, per indexer.</td>
</tr>
</tbody>
</table>

Adjusting the `auto_summarize.max_concurrent` setting in `savedsearches.conf` to 2 multiplies the IO, processing, and memory used while running scheduled acceleration searches on every indexer. A value of 2 provides the best performance increase, with higher values succumbing to diminishing returns. For configuration details, see Use parallel summarization to speed up creation and maintenance of report summaries in the Splunk Enterprise Knowledge Manager Manual.

Splunk administrators can use the monitoring console to monitor and track indexer resource use. For more details, see About the monitoring console in Monitoring Splunk Enterprise.

**Index parallelization**

Index parallelization allows an indexer to maintain multiple pipeline sets. A pipeline set handles the processing of data, from receiving streams of events, through event processing, and writing the events to disk. By allowing an indexer to create and operate multiple pipelines, multiple data streams can be processed with additional CPU cores, accelerating data parsing and disk writing up to the
limits of the indexer's I/O capacity. Customers leveraging index parallelization can see an increase in an indexer's sustained indexing load, or a doubling of indexing speed when receiving a sudden surge of data from the forwarders.

<table>
<thead>
<tr>
<th>Setting name</th>
<th>Default</th>
<th>Maximum recommended value</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>parallelIngestionPipelines</td>
<td>1</td>
<td>2</td>
<td>Multiplies the number of pipelines per indexer.</td>
</tr>
</tbody>
</table>

Adjusting the `parallelIngestionPipelines` setting in `server.conf` to 2 will use an additional 4-6 CPU cores, and requires 300-400 IOPS to maintain indexing throughput on every indexer. Also, there are fewer CPU cores available for search processing. A value of 2 provides the best performance increase, with higher values succumbing to diminishing returns. For configuration details, see Manage pipeline sets for index parallelization in the Splunk Enterprise *Managing Indexers and Clusters of Indexers Manual*.

Splunk administrators can use the monitoring console to monitor and track indexer resource use. For more details, see About the monitoring console in *Monitoring Splunk Enterprise*. 

34