Logger Configuration and Tuning: Best Practices

Logger Configuration and Tuning: Best Practices

About this Tech Note

The information in this technical note applies to all HP ArcSight Logger 5.5 appliance and software models, except where specifically noted.

This technical note does not provide scientifically calculated numbers but best practices obtained from existing customers, field, and HP ArcSight development and QA groups. It identifies and describes the Logger configuration components that can influence its performance and provides recommendations for obtaining optimal performance on a Logger system.

Additional guidelines and instructions are included in the applicable sections of the Logger 5.5 Administrator's guide, available from the ArcSight Product Documentation community at https://protect724.hp.com.)

Overview

HP ArcSight Logger (Logger) is a log management solution that is optimized for extremely high event throughput, efficient long-term storage, and rapid data analysis. Logger receives and stores events; supports search, retrieval, and reporting; and can optionally forward selected events.

Even though Logger is built for fast event insertion and forwarding, and high performance search and analysis, a combination of these activities occurring at the same time on Logger can cause Logger components to compete for resources and thus affect Logger's performance. The performance of your Logger system also depends on the network environment, the complexity of the functions you are performing on Logger, the Logger type, and how you have Logger configured.

Many factors can affect Logger's search speed and scan rate. The amount of time a search requires depends on, among other things, the size of the data set to be searched, the complexity of the query, and whether the search is distributed across peers.

When deploying and configuring Logger or troubleshooting it to achieve optimum performance, follow the guidelines discussed in this technical note. If you need additional guidance, contact HP Customer Support at http://support.openview.hp.com.

Input and Output Components

The following sections discuss factors to consider and provide guidelines for configuring Logger input and output components:

Web Connections

Logger supports up to 250 simultaneous HTTPS connections. These connections can be from Web browsers connecting to the Logger Web UI, from connectors to SmartMessage receivers configured on the Logger, from API clients, and from peer Loggers.

To determine how many connections to your Logger there are currently, run the following Linux command:

```
netstat –tlnp | grep <port>
```

where <port> is the port your users use to connect to the UI. This result includes peer searches and SmartConnectors.

To determine the number of Apache processes currently running on your system, run the following Linux command:

```
ps aux |grep httpd
```

Connectors

Connectors send events to Logger and can vary greatly in their peak throughput. Simple connectors sending smaller events, such as Cisco PIX Syslog, typically have higher throughput than more complex connectors with larger events, such as Windows Unified Connector events.

Logger supports up to 250 simultaneous HTTPS connections. Contact support if you need to increase this number. If you have a large number of connectors connecting individually to SmartMessage receivers on Logger, consider aggregating connectors.

Run a search like the following to get a list of connectors connecting to Logger.

```
...| top agentHostName
```

Receivers

Receivers listen for events. There is no imposed limit on the number or type of receivers, or the maximum throughput a receiver can handle. However, HP ArcSight recommends a maximum of 40 to 50 receivers, based on field observations. A high incoming event rate and large event size can affect the performance of a receiver. Additionally, the connectors that send events to the Logger may have limits on their throughput.

Use an individual receiver for each connector sending events to Logger, instead of using a single receiver for all connectors. This allows for better granularity in searches and the ability to monitor event flow for each receiver through alerts.

As with other considerations related to scope, questions about how to configure receivers and how many connectors should send data to any given receiver, are best answered when taking the entire environment into consideration (data type, usage requirements, and so on). The ArcSight Professional Services team can do full scoping of such scenarios. Contact your local ArcSight Sales representative for more details.

Devices

A device is a named event source, comprising of an IP address or hostname of the event sender and the name of the receiver that receives the event. Therefore, a host or connector that sends events to two different receivers on the same Logger is recognized as two different devices.

Device Groups

Device groups classify events received from various devices. For example, Device A and Device B events could be stored in device group AB and Device C events could be stored in device group C. There is no limit on the number of device groups on a Logger.

You can write storage rules that direct events from specific device groups to storage groups. Also, you can include device groups in queries to limit the data set that Logger must scan, thus resulting in the faster searches.

Forwarders

Forwarders send events received on Logger to specific destinations such as ESM, other connectors, or other Loggers. Logger uses its onboard connector when forwarding events to ESM. You can forward all events or specify filters to forward only specific events.

The rate at which a forwarder forwards events depends on a number of factors including the number of forwarders, the size of the events, and the complexity of the query used to filter the events. When no filter is used, (all events are forwarded) a single forwarder can forward up to 3K EPS to an ESM destination and 6K EPS to a syslog destination. The larger the event size, the lower the EPS rate.

When filtering events for forwarding, Logger must evaluate each event against the query to determine whether to forward it to the destination. This slows down the forwarding rate. The more complex the query is, the slower the forwarding rate. (A complex query typically includes a number of Boolean expressions, such as a regular expression with multiple OR operators.)

Sending CEF events to a Syslog destination from Logger can increase the outbound EPS rate up to 5-6K EPS. All other encrypted destinations have the 3K EPS limit.

Improving Forwarder Performance

The following guidelines can help optimize the forwarding rate.

When configuring forwarders:

- Reduce EPS in rate (into the receivers) through filtering and aggregation.
- Increase the cache for the forwarder to 10G. This will help prevent dropped events if the destination is down.
- Additional forwarders in Logger may increase EPS throughput. If you are using one forwarder, add a second one. Adding a second forwarder could increase the forwarding rate by 20-30%. HP has observed that when you add a second forwarder, the aggregate forwarding rate to ESM is approximately 3.2K EPS.
- Do not have more than ten forwarders on a Logger. Even though each additional forwarder improves the forwarding rate, the relation is not proportional. In high EPS situations or situations where other resource-intensive features are running in parallel (alerts, reports, and several search operations) and the forwarding filter is complex, having too many forwarders may reduce performance because forwarders have to compete for the same Logger resources besides competing for the onboard connector for forwarding.
- Instead of an additional forwarder, consider adding another Logger to distribute the forwarding load.
- Ensure that the forwarder's destination can keep up with the forwarded events. If it cannot, add another forwarding destination. For example, if an ArcSight Manager cannot keep up with the forwarded events, add another ArcSight Manager.
- To increase the outbound EPS limit if your forwarder has no filters, try adding a second logical ESM destination. This can help increase the limit but may have some performance impact on the Logger.
- To increase the forwarding rate to an ESM destination, create a secondary ESM destination with a secondary forwarder.
- To increase the outbound EPS limit for forwarders with filters, move the filtering operation from the Logger forwarder to the source connectors and devices. Doing so removes the need to filter events, and you can then forward all events.

When forwarding to ESM:

- Disable event aggregation (from the ArcSight Manager).
- Make sure the "preserve raw events" is turned off for the connector (Logger's Forwarder connector in ESM). This is also set at the ESM destination.
- Disable real-time alerts on Logger and use rules/alerts within ESM instead.

- Disable basic aggregation for Logger's forwarding connector because it is resource intensive. Basic aggregation is set in the ArcSight Console.
- Disable DNS lookup on the Forwarder connector in ESM.
- Use one forwarder and apply a filter-out filter on the connector resource in ESM to exclude data that you do not want to forward.
- While adding additional forwarders can increase EPS throughput to ESM, configure only one ESM destination for each ESM server. Each additional ESM destination shares memory with all configured ESM destinations, which can cause contention and potential connector failure if oversubscribed. As a workaround, you can increase the Logger on-board connector from 256MB to 512MB from the ESM console, or logically separate the events into several active channels once they arrive to the ESM.
- When separating incoming filtered events on ESM, use Active Channel filters instead of creating multiple Active Channels on multiple incoming Logger connectors.
- Separate the events from the source connectors into two streams, each one of them going to a dedicated receiver. Use one stream for the events that need to be forwarded to ESM and the other stream for the events that do not need to be forwarded. Then, define a filter condition on the device or device group receiving the events from the first stream. Doing so enables you to configure an efficient filter condition.

When writing forwarder queries:

- Follow the guidelines for query performance and scan rate provided in “Improving Search Performance” on page 10 when writing queries for forwarders.
- When filtering events in Regular Expression queries, use the meta-data query terms `_storageGroup` and `_deviceGroup`. Including storage groups in searches is more efficient than including device groups.
- Make queries as simple as possible. Simplify regex filters at the forwarders.
- Use Unified Query forwarders as much as possible. In most cases, Unified Query based forwarding is faster than Regular Expression based forwarding. Convert Regex filters to Unified filters where possible.
- If you want to forward all events that have the same data at the beginning or end of an event, anchor the regular expression in the forwarding filter for efficient filtering. For example, if you want to forward all events that start with "CEF", use "^CEF" in the regular expression instead of "CEF" because "^CEF" will match the first three characters of the event, and if a match is found, the event will be forwarded. If you use "CEF" in the query, Logger will scan the entire event for the string “CEF”.

Storage Components

The following sections discuss factors to consider and provide guidelines for configuring Logger storage components:

Storage Volume

Storage volume defines Logger's primary storage space. Although you can increase the size of an initially defined storage volume, follow these guidelines for optimal use of available storage space and expected performance.

Do not use NFS as primary storage. Although this setup is possible, HP ArcSight does not recommend it due to sub-optimal performance and reliability. You can use NFS for Archive storage.

- You can increase the size of a Storage Volume, but not decrease it. Each Logger model has a maximum allowed Storage Volume size.
- On a SAN Logger appliance, make sure that you allocate the maximum size LUN during initial Logger setup. Logger cannot detect a resized LUN. Therefore, if you change the LUN size after it has been mounted on a Logger, Logger may not recognize the new size.

Storage Groups

Storage groups enable you to implement different retention policies. Therefore, data stored in one storage group can be held for longer or shorter time than another group. .

You cannot rename storage groups after you create them.

In many cases, storage group retention policies are dictated by compliance requirements, such as PCI. However, such requirements might not be met if the storage groups fill up, because the oldest events could be purged automatically to make room for incoming events, even if they are still within the retention period. Even though you set the Default Storage group to 365 days retention, you should not simply assume that all the data will still be there on day 365 in a growing environment. As your environment grows, it is important re-scope your requirements for receivers, forwarders, retention policies, number of Loggers, and so on, accordingly. The ArcSight Professional Services team can do full scoping of such scenarios. Contact your ArcSight Sales representative for more details.

To ensure better control of storage group retention and disk space utilization, do not allow your storage group utilization to increase above 90%. As storage groups near 99% utilization, they start running out of disk space, which reduces the performance of searches due to increasing fragmentation.

Configure alerts to notify the appropriate users when the Storage Group usage gets too high. For more information, see “Notifications” on page 7, “Disk Space and Database Fragmentation” on page 9, and the Logger Administrator’s guide.

Storage Rules

Storage rules direct events from specified device groups to specific storage groups. Use storage rules to direct events to the correct storage group. For example, you could set up storage rules to store events from specific sources in storage groups that have different retention periods. You can create up to 40 storage rules.

Notifications

You can set up alerts to be triggered by specified events or event patterns and, optionally, to send notifications to previously configured destinations such as email addresses or SNMP servers. Logger provides two types of alerts, real-time alerts and saved search alerts.

Sending alerts via email is controlled by the SMTP server settings under System Administration. However, the Report Engine has its own SMTP server settings. For SMTP email functionality across Logger, be sure to configure it in both places.

The following sections discuss factors to consider and provide guidelines for configuring alerts on Logger:

- "Real-time Alerts" on page 7
- "Saved Search Alerts" on page 7

Real-time Alerts

Alerts are triggered in real time. That is, when a specified number of matches occur within the specified threshold, an alert is *immediately triggered*. Although any number of real-time alerts can be defined, only five can be enabled at any time.

You can use preconfigured filters to specify event patterns when creating alerts.

Save a copy of a preconfigured filter and edit the copy to meet your business needs (or just write your own.) Refer to the Logger Administrator's guide for more information.

The particular filters available depend on your Logger version and model, but may include:

- System Alert - Disk Space Below 10% (CEF format)
- System Alert - Root Partition Free Space Below 10% (CEF format)
- System Alert - Storage Group Usage Above 90% (CEF format)

Use the system filters for the five real-time alerts to quickly find and handle system or hardware issues. Create saved-search alerts for other things, such as log source alerts.

Real-time alerts can affect system performance, especially if many other resource-intensive features are running on Logger in parallel.

Saved Search Alerts

Scheduled Search/Alerts (saved search alerts) are triggered at scheduled intervals. That is, when a specified number of matches occur within the specified threshold, an alert is triggered at the next scheduled time interval. For example, if a saved search alert is set to trigger every hour when five matches occur in sixty seconds and if five matches occur between 12:05 PM to 12:06 PM, the alert will be triggered at 1:00 PM. Refer to the Logger Administrator's guide for more information on alert triggers and notifications.

Although you can define any number of saved search alerts, a maximum of 50 can run concurrently. Contact support if you need to change this number.

To ensure system performance, a maximum of 200 alerts are allowed per saved search alert job. Therefore, if a saved search alert job triggers more than 200 alerts, only the first

200 alerts are sent out for that job iteration; the rest are not sent. Additionally, the job is aborted so it does not trigger more alerts for that iteration and the status for that job is marked "Failed" in the Finished Tasks tab (**Configuration | Scheduled Tasks > Finished Tasks**.) The job runs as scheduled at the next scheduled interval and alerts are sent out until the maximum limit is reached.

Time Settings and Scheduled Tasks

Precise time stamping of events is critical for accurate and reliable log management. The times displayed for Logger operations such as searches, reports, and scheduled jobs are in the Logger's local time zone.

Follow these guidelines to ensure accuracy of time and optimal scheduled task handling:

- Use an NTP server instead of manually configuring time and date on your system.
- Avoid scheduling tasks to run during the hour that is lost or gained at the start and end of the daylight saving time period (DST).

  Scheduled operations such as reports, event archives, and file transfers are affected when system time is adjusted on the Logger at the start and end of the DST.

  - Operations scheduled for the hour lost at the start of DST (in early spring) will not run on the day of time adjustment.
  - Operations scheduled for the hour gained at the end of the DST (in late fall) will run at standard time instead of the DST time.

Event Archives

Event Archives enable you to save the events for any day in the past, not including the current day. When events are archived, index information for those events is not archived. Therefore, when event archives are loaded, indices are not available. As a result, a search query that runs on archived events that have been loaded on Logger is slower than when the data was not archived because the index data for the archived data is not available.

The primary function of event archives is to allow for long-term storage of events that are not stored locally on Logger (outside any storage group retention policy). An added advantage is that in the event of total data loss, such as in the case of appliance failure, any data that is archived will still be accessible over NFS/CIFS. This does not, however, fulfill the requirements for full disaster recovery because event archives do not contain indexes and therefore, searches and reports, run on archive data, will run much more slowly, possibly timing out. Refer to the Logger Administrator's guide for information on how to increase the client time out and the database connection timeout.

For full disaster recovery planning, consider having multiple Loggers in a High Availability setup with events being dual-fed from the connectors. The ArcSight Professional Services team can do full scoping of such scenarios. Contact your local ArcSight Sales representative for more details.

Do not move archived files from versions earlier than Logger 5.1. If moved from their original location, archives from earlier versions cannot be loaded on to the Logger. If you need to delete the archives, use the Logger user interface to do so (**System Admin |Storage Remote File Systems**.) Any attempt to load or delete an old archive will look for the original remote archive location. If this was deleted it will need to be added back again with the same name, even if the archive itself was moved to another server.

Follow these guidelines for optimal performance when archiving events:

- Archive during off-peak hours.
- Only one manual archive job can run at a time. However, a scheduled archiving operation can run in parallel with a manual job.
- For a manual archive operation, do not archive too many days or storage groups worth of data at a time. If you have a large data set to archive, archive in smaller chunks to prevent a negative impact on Logger's performance.
- Ensure daily scheduled Event Archives are accompanied by daily scheduled Configuration Backups. Without a daily Configuration Backup, Event Archives from the previous day will not be usable in the event of total data loss. In such scenarios, a restore of a previous configuration backup from an earlier week/month will only allow access to event archives up until that point.

Restoring Archives

Events are not copied back to local storage when event archives are loaded. Instead, a pointer to the archive is activated and it is included in queries.

While there is no limit to how many archives can be loaded, as the number of loaded archives increases, the size of the metadata table that tracks the data increases, which makes the queries slower. If you load a large number of archives, searches on the regular data may be slower. How much slower, depends on how much data is in the archives and also on how much regular indexed data is in the system.

It might be worthwhile to try restoring the Event Archives on a freshly installed Logger that has had a Configuration Backup applied as long the same archive mount names are attached the new Logger.

Disk Space and Database Fragmentation

Sufficient disk space on Logger is important for all functionality to work correctly. It is important to ensure that at least 50% of the root disk (/) is free for usage by the system as and when needed.

Do not confuse disk space usage under the root disk (/) with usage under /opt/data where events are stored. The area under /opt/data is always 100% full when pre-allocation is configured during initialization.

As the Logger database expands, more indexing is required and there are more events to scan. This can result in decreased search speed. To help maintain and improve search speed as your database grows, you need to perform database fragmentation frequently.

Configure alerts to notify the appropriate users when the free space gets too low and when the Storage Group usage gets too high, and defragment the database at those times. See "Notifications" on page 7 and the Logger Administrator's guide for more information.

Global Summary Persistence

The Global Summary Persistence functionality is disabled in this release. As soon as possible after upgrading to Logger 5.3 SP1, enter system maintenance mode and defragment the Global Summary table. If you defragmented the table after upgrading to

5.3 SP1, you do not need to repeat the procedure upon upgrade to Logger 5.5. Refer to the Logger 5.5 Administrator's Guide for instructions.

Search

The following sections discuss factors to consider and provide guidelines for searching and search performance:

Exporting Search Results

Logger can return a maximum of 1M matching records for any search operation and export up to 1 million records from the search results. The performance of an export operation depends on the size of the search results data set. When a very large set of search results is exported, you may observe sub-optimal export performance.

Indexing

For faster searching, index all fields you use in queries. To see which fields will be indexed on your system, open the **Configuration | Search > Default Fields** page and look for the checkmark  in the **Indexed** column for the field.

Significantly exceeding HP ArcSight's default recommended indexed fields could result in performance degradation in certain situations. Only index those additional fields that are necessary for your environment. Once a field has been added to the index, you cannot remove it.

Allow time between adding a field to the index and using it in the search query. If Logger is in the process of indexing a field and you use that field to run a search query, the search performance for that operation will be slower than expected.

Improving Search Performance

Many factors can affect search speed and scan rate. The amount of time a search requires depends on, among other things, the size of the data set to be searched, the complexity of the query, and whether the search is distributed across peers.

No two Loggers are the same. Even when the version, hardware model, platform, and configuration is the same, the data is different, and the load upon each system varies greatly from moment to moment, so there is no single "right" value for query or forwarding speed.

The following guidelines can help optimize search performance.

How the System is Set Up

How you set up your environment and organize your data can affect search performance. For optimal performance:

- Have a fast network.
- Configure peer Loggers on the same subnet.
- Partition the data so that it can be searched in chunks rather than all at once.
  - Use peers to distribute the data.
  - Use storage groups to divide the data.

High Event Input

When the event input is high, indexing can lag behind. As result, the search defaults to a slower non-indexed search.

- To avoid this issue, run a fixed time search that does not to include the last two minutes.
- If this is a recurring problem, make sure that your environment is sized correctly. The ArcSight Professional Services team can do full scoping of such scenarios. Contact your local ArcSight Sales representative for more details.
- On the Connector, turn aggregation on to lower the number of duplicate events. This will also lower down the EPS rate.
- Use the Search Analyzer tool to determine if your data is fully indexed. See the Logger Administrator's Guide for details.

For a quick check to see if your index is up to date, the Global Summary includes the date and time of the most recently indexed data.

The Number of Events That Must be Scanned

Restricting searches to specific storage groups or peers decreases the number of events to search because storage group or peer filter is applied before the query is executed. If there are fewer events to scan, as is usually the case when looking at a single storage group rather than all of them, the result returns more quickly.

Use metadata query terms `_storageGroup` and `_peerLogger` to limit the number of events that must be scanned.

Including storage groups and peers in search queries is more efficient than including device groups. Use storage groups and peers in the query as much as possible, to reduce the amount of data searched.

- To limit the search to the Logger at `192.0.2.9`, use the following:

  ```
  _peerLogger IN [192.0.2.9]
  ```

- To limit the search to the default storage group, use the following:

  ```
  _storageGroup IN [Default Storage Group]
  ```

Search Timeframe

Searching against a longer timeframe takes longer than searching against a shorter timeframe, since there are more events to search. For faster results, limit the search to a shorter timeframe.

The Number of Events that Match the Search

Searches that result in a high number of matching events will be slower than searches with lower event match. For example, searches with more than 1 million matching events will be slower than search with few thousand matching events, since there will be fewer events to load in the memory.

- If the search results returns with large number of matches, modify the search query to make it more specific. For example, instead of `authentication | where name CONTAINS "failure"` use the following.

  ```
  authentication AND name CONTAINS "failure"
  ```

- Use metadata query term `_deviceGroup` to reduce the result set to certain device groups for each search condition. For example, to limit the search to the smart device group on the Logger at 192.0.2.9, use the following.

  ```
  _deviceGroup IN [192.0.2.9 [smart]]
  ```

- Where possible, write queries that take advantage of superindexes. For information on how to write super-indexed field queries optimally, including examples, consult the Logger Administrator's guide.

Using Regular Expression Queries

Regular expressions do not utilize indexing, so queries containing regular expressions (and search operators that result in regular expression-type parsing, such as REX) can make search operations slow. To optimize search speed when using regular expressions in queries, make sure the data set that the regular expression needs to scan is small.

To control the data set:

- Precede the regular expression with a search term that reduces the data set size. For example, to extract the IP address from all events that contain the words "telnet" and "failed", use these words as the full-text search terms to reduce the data set that the following regular expression will need to scan:

  ```
  telnet failed|regex ="(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})"
  ```

- Use metadata such as device group, storage group, and peers instead of Boolean operators to filter events, where possible.

Including storage groups and peers in search queries is more efficient than including device groups. Use storage groups and peers in the query as much as possible, to reduce the amount of data searched.

Using Boolean Operators in the Query

Search speed can vary depending on the search conditions used. A query that includes OR or AND operators takes longer to process. The OR operator is particularly resource intensive because it requires the regular expression to scan the text of each event multiple times. To determine if this is happening, reduce the number of OR and AND operators and run the searches again.

Using AND and OR with the = operator can be very powerful when searching super-indexed fields. However, to obtain the greatest search speed improvement, you must use them carefully. For information on how to write super-indexed field queries optimally, including examples, consult the Logger Administrator's guide.

The Complexity of the Query

The search speed can vary depending on the query's complexity. Reduce the complexity of your queries where possible. Use simple operators like `replace` and `rename` and reduce the use of complex operators such as `rex`, `sort`, and `chart`.

Whether All Fields in the Query are Indexed

- Searches where all fields are indexed are faster than searches with non-indexed fields. Use queries where all fields are indexed as much as possible. A list of the default fields, along with their index status is available on the Default Fields tab (**Configuration | Search > Default Fields**.)
- To speed up a non-indexed search, combine index field based search or full text search with non-indexed field search. For example, since `requestUrl` is not an indexable field, instead of `"requestUrl CONTAINS "username""`, use the following.

```
name = "TCP_MISS" | where requestUrl CONTAINS "username"
```

Even though a search query includes only indexed fields, you might not realize the performance gain you expect in these situations:

- When you perform search on data in a time range in which a currently indexed field (included in the query) was non-indexed, the query will run at the speed you would expect if the field was not indexed. This is because new indexing information is not applied to previously stored events. For example, you index the "port" field on August 13th at 2:00 PM. You run a search on August 14th at 1:00 PM. to find events that include port 80 and occurred between August 11th and August 12th. The "port" field was not indexed between August 11th and the 12th. As result, the search defaults to a slower non-indexed search.
- When a query that includes indexed field is performed on archived events, the query runs slower than when the data was not archived. This occurs because the index data is not archived with events. As result, the search defaults to a slower non-indexed search.
- When you include a field in your search query that Logger is in the process of indexing, the query will run slowly. This issue is discussed in "High Event Input" on page 11.

Whether the Query takes Full Advantage of Super-Indexed Fields

When you need to search for uncommon values in the following IP address, host name, and user name fields, take advantage of superindexing for faster search speeds. Superindexes rule out chunks of data from your search and return search results very quickly when there are few or no hits.

Table 1 Fields With SuperIndexes

destinationAddress	destinationHostName	destinationPort
destinationUserId	destinationUserName	deviceAddress
deviceEventClassId	deviceHostName	deviceProduct
deviceVendor	sourceAddress	sourceHostName
sourcePort	sourceUserId	sourceUserName

Search on super-indexed fields only using the = operator, and only AND with non-super-indexed fields for fastest search performance. For more information on how to write super-indexed field queries optimally, including examples, consult the Logger Administrator's guide.

The Number of Concurrent Searches, Reports, and Forwarders

Searches, forwarders, and reports all use the same search engine. When there is a heavy load on the system, such as a high incoming EPS, forwarding with filtering, and multiple search and report operations going on in parallel, it will take longer to execute a query.

Spread resource-intensive tasks to off-peak hours as much as possible. Schedule searches and reports to run at a time when there is not much load on the system or reduce the load when your searches or reports need to run.

The Size and Type of Events

Searches against small size events, such as syslog (where the event size varies from 1K-1.5K) will be faster than events with larger size such as Blue Coat events (where the event size varies from 2.5K-4K). This behavior will be more noticeable when the search is a non-indexed search.

Logger Options that Affect Search

If the Discovered Field and Summary Field options are enabled, the system will try to populate these fields during the search, which can slow it. This becomes more noticeable when there is high event match. For faster results, disable these options.

Other options that might affect search speed include:

- Secondary Delimiter Support—Turn it off to improve performance (**Configuration | Search > Search Options**)
- Source type support—Use specific source types to improve performance.
- Global Summary Persistence—Defragment the table after upgrading to 5.3 SP1.

Other Factors that Can Affect Search Speed

- Logger version
- Appliance and model

- The number of events already in the system
- Ingestion rate (insertion rate)

Peer Loggers

The following sections discuss factors to consider and provide guidelines for configuring and searching across peers:

- "Authentication" on page 15
- "Using the CEF Search Operator" on page 15
- "Improving Peer Search Performance" on page 15

Refer to the Logger Administrator's guide for the number of peers the can be configured.

Performance issues in peering operations may indicate the need for Global Summary Persistence defragmentation. For instructions, refer to the Global Summary Persistence Defragmentation section of the Logger Administrator's Guide.

Authentication

For security reasons, HP ArcSight recommends that you use authorization IDs to establish peer relationships.

- If the remote Logger is configured for SSL Client authentication (CAC), you must configure an authorization ID and code on the initiator Logger.
- If user name and password are used for authenticating to a remote peer Logger, the credentials are only used one time, during the peering relationship set up. After a relationship has been established, the credentials are not saved (on the Peer Loggers page) and the peers do not authenticate periodically. Therefore, if the user name or password used to establish a relationship is changed at a later date or the user name is deleted, peering relationship is not broken. However, if you delete the peering relationship or it breaks for other reasons, you will need to enter the updated credentials to re-establish the relationship.

Using the CEF Search Operator

With Logger versions 5.2 and later, you do not need to explicitly extract the CEF fields and then apply other search operators to those fields. The CEF operator is implicit. You can specify the event fields directly in queries. For example, to find the top values in the message field, instead of `...| cef message | top message`, use the following:

```
...| top message
```

When you run a peer search, initiate queries that do not explicitly use the CEF operator from a Logger running version 5.2 or later. A query that does not use CEF defined fields will run if the query is initiated on a Logger running version 5.2 or later. However, if the query is initiated on a 5.1 or earlier Logger version (before CEF was deprecated), it will fail. For more information, see "Improving Search Performance" on page 10.

Improving Peer Search Performance

Searches done across peer Loggers are done locally on the peer rather than the Logger initiating the search. The following guidelines can help optimize optimize the performance of peer searches.

When configuring peers:

- If you set up a number of peers on a local network to horizontally scale out the system, be sure to configure them identically.
- If you added custom schema fields to your Logger schema, those fields must exist on all peers. Otherwise, a search query containing those fields will not run (when run across peers) and return an error.
- The time and date of the system on which the software Logger is installed must be set correctly with respect to its time zone to peer with other Loggers. HP ArcSight recommends that you configure the Logger system to synchronize its time with an NTP server regularly.

When running searches across peers:

- Follow the guidelines for query performance and scan rate provided in "Improving Search Performance" on page 10 when writing queries for searches across peers.
- Ensure that the device and storage groups specified in the query exist on all peers. Peers on which a device or storage group does not exist are skipped.
- Make sure that event fields on ALL peers are indexed for the time range specified in a query. If an event field is indexed on a local Logger but not on its peers for a specific time range, a distributed search will run at optimal speed on the local Logger, however will run slower on the peer Loggers. Therefore, the search performance in such a setup will be slow.
- For peers with different schema, make sure that your searches and reports only involve fields that have the same name and data type on all peers. Otherwise, the search or report will fail.
- When peers of mixed Logger versions are involved in the same search, the search features you can use are determined by capabilities of the peer with the earliest, and therefore most limited, version. For example:
  - If the earliest peer Logger is version 4.0, you can use full text (keyword) search, search operators, and histograms.
  - If the earliest peer Logger is version 5.1, you can use full text (keyword) search, search operators, and histograms.
  - If the earliest peer Logger is version 5.2, you can use full text (keyword) search, search operators, histograms, and custom schema.
  - If the earliest peer Logger is version 5.3, you can use full text (keyword) search, search operators, histograms, custom schema, and source types with parsers.
  - If the earliest peer Logger is version 5.5, you can use full text (keyword) search, search operators, histograms, custom schema, source types with parsers, and super-indexed fields.

For details of available capabilities, such as available search operators, refer to the release notes of the earliest peer Logger.

Reports

Reports that must process very large data sets can be resource intensive. HP ArcSight recommends running scheduled reports instead of ad hoc reports whenever possible, so that most reports are run during periods of light load.

When isolating a specific Report or Report folder to users, the Report rights must include all associated Report Folders rights within the folder structure tree. For example, if Report Folder: Anti-Virus is the target report folder, you must also include the rights to Report Folder: Device Monitoring. Refer to the Logger Administrator's Guide for instructions.

The following sections discuss factors to consider and provide guidelines for reporting:

Improving Report Performance

The following guidelines can help optimize report performance.

When running reports:

- Run no more than 10 scheduled concurrent reports.
- Ensure that published reports and saved searches are kept ONLY for as long as required, especially when run ad-hoc, as they take up disk space resources. For example, running 10 reports that each generate 1 GB files will utilize 10 GB of space that could otherwise be used by the system. Low disk space under / (root on Appliances) or $ARCSIGHT_HOME (Software Logger installation path) can be a result of large and potentially unnecessary (or unused) CSV exports and published reports.

When scheduling published reports, HP ArcSight recommends that you change the retention period to 1 week after generation. To do this, use the **Valid Upto** <*N*> <*Unit of time*> **After Generation** option on the Add Report Job page.

- Running large reports can take up a lot of space temporarily, which could cause the report to fail, if space is limited.

Configure alerts to notify the appropriate users when the free space gets too low. For more information, see "Notifications" on page 7, "Disk Space and Database Fragmentation" on page 9, and the Logger Administrator's guide.

- If your reports contain millions of events, contact Customer Support to increase the heap size.
- Specify a scan limit for reports run manually. The default scan limit is zero, which means all events. When you specify a scan limit, the latest N events are scanned. This results in faster report generation and is beneficial when you want to process only the latest events in the specified time range instead of all the events stored in Logger.
- In addition to the search fields, all fields displayed in the report should be indexed. In addition to the fields in the WHERE clause of the query, the fields in the SELECT clause also need to be indexed. A list of the default fields, along with their index status is available on the Default Fields tab (**Configuration | Search > Default Fields**).

When writing report queries:

- Follow the guidelines for query performance and scan rate provided in "Improving Search Performance" on page 10 when writing queries for reports.
- Use the where clause to specify conditions to narrow down your results, for example you could use queries like the following:
  `where <fieldName>="<value>"`
- Select specific fields, avoid patterns that will return too many hits.
  - Use queries like the following:
    `Select <fieldName1>, <fieldName2> … from events`

- Avoid queries like the following:
  `Select * from <tableName>.`
- For large reports, add a filter like the following, to the SQL, before the sort and order condition:

```
select events.arc_sourceAddress,
events.arc_destinationAddress,
events.arc_destinationPort,
events.entTime,
from events where events.arc_destinationPort >22 and
events.arc_categoryOutcome="/Failure" ;
```

- Avoid aggregation operations over large data sets.
- Avoid self-joins.
- Avoid sub-queries such as `Select * from <tableName> where <fieldName> in (select …)`
- Be aware of the following when using order by:
  - Avoid using order by with a large amount of data, as that will take a long time.
  - Limit the number of rows you want to order by.
    - Use queries like the following:
      `Select ... from <tableName> group by <fieldName> order by <fieldName>`
    - Avoid queries like the following:
      `Select ... from <tableName> order by <fieldName>`
  - Avoid sorting on entire fields that are very large, because that will use a lot of disk space. Use a substring of a field instead of the full length of the field if the substring is good enough.

    In the examples below, we use the name field, which is 512 char long.
    - Use queries like the following:
      `SELECT * FROM <tableName> order by substr(name,1, 64) LIMIT 50;`
    - Avoid queries like the following:
      `SELECT * FROM <tableName> order by name LIMIT 50;`
- Be aware of the following when using group by:
  - Group by contains order by.
    - If you do not need order by, do not include it in the query.
    - If you need order by, change the query to order by a short field instead of a long one. (Group by also uses the same fields as order by.)
  - Limit the number of rows to sort.
    - Use queries like the following:
      `Select ... from <tableName> group by <fieldName> order by <fieldName>`

      or
      `Select ... from <tableName> where <fieldName>= .... group by <fieldName> order by <fieldName>`
    - Avoid queries like the following:
      `Select ... from <tableName> order by <fieldName>`

- Avoid using order by directly on the event table.

  Use queries like the following:
  ```
  Select ... from <tableName> where <fieldName> =...  order
  by <fieldName>
  ```

Report Timeout Settings

If your report is timing out, you can increase the DATABASE_TIMEOUT and the HTML_VIEWER_TIMEOUT. However, increasing the Report Timeout above the default setting of four hours puts additional load on the system because spacing out of reports over any given day becomes more difficult, particularly since manual reports and searches compete for resources. Refer to the Logger Administrator's guide for information about timeouts that can affect long running reports.

Although you can increase the default timeout settings for scheduled reports, HP ArcSight recommends that you optimize the report query instead. For example, if you do not need all the events or too many will be returned, you can get a sample by using a scan limit. When you specify a scan limit, the latest N events are scanned. The default scan limit is zero, which means all events.

Improving Performance of Distributed Reports

Distributed reports include matching events from the specified peers of the originating Logger.

Use the following guidelines to help optimize the performance of distributed reports:

- Avoid running a distributed report on a Wide Area Network (WAN) link.
- Avoid running more than 3 concurrent distributed reports.
- When writing queries for distributed reports, follow the guidelines for query performance and scan rate provided in "Improving Search Performance" on page 10.
- If you are running the report on a very large data set and the performance of the report is not optimal, reduce the size of the data set.
- All Loggers on which you are running the distributed report must be running Logger 5.2 or later.
- Use pushed functions in WHERE clauses

  SQL functions that are pushed to peers include:

  - String functions
    ```
    char_length, char, concat, insert, lcase, left, length,
    locate, lpad, ltrim, replace, right, rpad, rtrim, strcmp,
    substr, trim, ucase
    ```
  - Numeric functions
    ```
    abs, ceiling, floor, round, sign, truncate
    ```
  - Date/time functions
    ```
    cast, dayofmonth, hour, minute, month, second, str_to_date,
    time_to_sec, unix_timestamp, year
    ```

iPackager Report Backup

Although the iPackager utility is primarily to allow quick distribution of reports to multiple Loggers, it can be used as a report backup tool. You can package all or selected reports and report objects residing on a Logger. This package can be later imported on a different Logger installation at any time.

When writing reports for export to other systems, use the default group, and device independent syntax, so the system content will not be over written.

Be sure to use device-independent syntax so that the report will run on other systems after distribution.

System Administration

A Logger Appliance with a failed hard drive will display a warning message. HP ArcSight strongly recommends that you contact support immediately to get the drive replaced.

Authentication

If you are using LDAP or RADIUS authentication, HP ArcSight strongly recommends configuring a backup LDAP/RADIUS server to help ensure uninterrupted access to Logger.

NICs

When setting the IP addresses for the network interface cards (NICs), hostname, and default gateway for your system, make sure that your DNS can resolve the host name you specify to your system's IP address. Performance is significantly affected if the DNS cannot resolve the host name. The Hostname in the CSR must be the same as the system host name.

User Groups and Search Group Filters

Implementing Logger users and groups to view only specific events can have performance implications, depending on the filters used to determine the events that the users can see.

- Use indexed search queries (Unified Queries) as much as possible. In most cases, unified query based searches are faster than regex based searches.
- To filter events in regular expression queries, use meta-data instead of Boolean operators as much as possible. Including storage groups and peers in searches is more efficient than including device groups. Use storage groups in the query as much as possible, to reduce the amount of data searched.

System Health Events

To monitor Logger's health and performance, review the system health events by using SNMP or Logger search. For more information, see "Notifications" on page 7 and the Logger Administrator's guide.

Web Services

The Logger Service Layer exposes Logger functionalities as Web services. By consuming the exposed Web services, you can integrate Logger functionality in your own applications. Using the Web service APIs, you can create programs that execute searches on stored Logger events or run Logger reports, and feed them back to your third-party system.

Using Special Characters in Regex Queries

To run queries such as | regex "," (or other special characters such as ì<>î) when doing a Logger search, turn on base64 encoding on the Logger side and use base64 decoding on the client side.

To turn on base64 encoding on the Logger side:

Add the following line in the /userdata/logger/user/logger/logger.properties file and to the /userdata/logger/user/logger/logger_webservices.properties file. (Create this property file if it does not exist.)

```
api.search.base64encode=true
```

To use base64 decoding on the client side:

Add the base64 decoding as shown in the highlighted location in the runSearch() method of the Web service client:

```
Tuple[] tuples = searchService.getNextTuples(...);

        for (Tuple tuple : tuples) {

                    String[] arr = tuple.getData();

                    for(int j=0; j< header.length; j++){

                                arr[j] = new
String(Base64.decode(arr[j])); // <= Add this line to decode the
received string using base 64.

                }

        }
```

Logger on Logger

Logger has several features that you can use to get more information about how Logger is doing and how it is being used.

Calculating Logger 5.5 Raw Events Size and Compression

The information here will help you determine the average raw event size for events collected from Smart Connectors.

Software Logger has a license that provides daily data usage and data usage enforcement. To see usage date on your Software Logger, log into the Logger UI and open **Configuration | License Information > Data volume Restrictions**.) This functionality is not available for Appliance Logger models, so the Logger appliance does not enforce daily data license usage.

You can use agent events stats to determine events size and event count for the data collected from Smart Connectors. You need to collect data for at least 24 hours (or more) to determine the daily data usage and the average raw events.

The compression rate can vary depending on several factors, such as the following:

- The size of events
- The type of events
- The uniqueness of fields in the events
- The EPS rate

Because of these and other variables, it is hard to predict the compression rate. The value differs from Logger to Logger. Based on what we have seen, the average compression rate

ranges from 8-10x.

Average Raw Event Size for Licensing

To calculate the daily data for the raw events, you can use the event information generated from the Smart Connector (deviceEventClassId =agent:050).

Use the following fields:

- **deviceCustomNumber3:** The number of non-internal events seen by this component since the last internal event.
- **deviceCustomString4:** The number of characters in the raw events in the non-internal events seen by this component since the last internal event.

To calculate the daily data for raw events:

Log into the Logger UI and open **Reports > New Query**. Save the following query. Then create and run a new report with the new query:

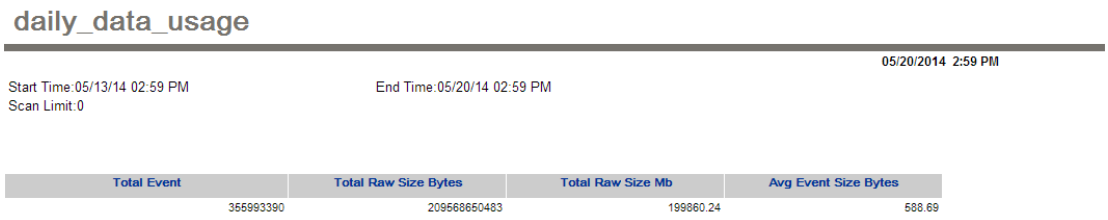
```
select sum(events.arc_deviceCustomNumber3) as "Total_event",
sum(events.arc_deviceCustomString4) as "Total_raw_size_bytes",

sum((events.arc_deviceCustomString4)/1048576) as
"Total_raw_size_MB",

(sum(events.arc_deviceCustomString4)/sum(events.arc_deviceCustomNu
mber3)) as "avg_event_size_bytes"

from events

where events.arc_deviceEventClassId = 'agent:050'
```



daily_data_usage

05/20/2014 2:59 PM

Start Time:05/13/14 02:59 PM End Time:05/20/14 02:59 PM
Scan Limit:0

Total Event	Total Raw Size Bytes	Total Raw Size Mb	Avg Event Size Bytes
355993390	209568650483	199860.24	588.69

This daily_data_usage report covered a 24-hour period and got the average raw event size and total event count and data usage.

To get a list for events per day over time:

Login to the Logger UI and open **Reports > New Query**. Save the following query. Then create and run a new report with the new query:

```
select DATE_FORMAT(events.arc_deviceReceiptTime,"%Y-%m-%e") as
"date",

sum(events.arc_deviceCustomNumber3) as "Total_event",
sum(events.arc_deviceCustomString4) as "Total_raw_size_bytes",

sum((events.arc_deviceCustomString4)/1048576) as
"Total_raw_size_MB",
```

```
(sum(events.arc_deviceCustomString4)/sum(events.arc_deviceCustomNu
mber3)) as "avg_event_size_bytes"

from events

where events.arc_deviceEventClassId = 'agent:050'

group by date
```

Data_usage/day

05/21/2014 11:48 AM

Start Time:Tue May 13 00:00:00 PDT 2014 End Time:Wed May 14 23:59:59 PDT 2014
Scan Limit:0

Date	Total Event	Total Raw Size Bytes	Total Raw Size Mb	Avg Event Size Bytes
2014-05-13	72078051	37055875753	35339.24	514.11
2014-05-14	77129190	38341716456	36565.51	497.11
2014-05-15	471215	291094059	277.61	617.75

September 8, 2014