Performance Guideline for syslog-ng Store Box

March 21, 2017

Abstract
Performance analysis of syslog-ng Store Box

BALABIT

CONTEXTUAL SECURITY INTELLIGENCE

Table of Contents

1. Preface

This whitepaper enables syslog-ng Store Box (SSB) end-users, integrators, and sales personnel to make predictions about the performance of the SSB appliance based on various environmental and configuration parameters.

The measured data refers to physical SSB appliances.

2. Log collecting performance

In this simple case we only measured sustained throughput of receiving logs in SSB. This means that there were no other operations like search ongoing at the time of the test. The source of messages was always TCP and we tested with and without encryption enabled on the source. The target logspace was using indexed, compressed log store where all fields were indexed using the default delimiters. We used the default 1024Mb memory limit. There was no pattern database loaded.

Factors with significant effect on performance

On physical hardware, SSB performance is usually limited by available processing power, that is, the performance is CPU bound.

During RAID synchronization the performance can drop significantly, because of the heavy load of the disk subsystem. As a result, a new SSB installation may give misleading performance numbers. Always wait for the RAID synchronization to finish before testing the performance.

If you use DNS for the log source, but the DNS server is unreachable, the performance of SSB will greatly decrease.

Sending logs from one log source into multiple logspaces degrades performance. If possible, send logs from one (or more) log source into a single logspace, and not duplicate the logs. If you need to filter or aggregate log messages in different ways, consider using the filtered logspace and multiple logspace features. (For details, see *Procedure 8.4, Creating filtered logspaces* in *The syslog-ng Store Box 4 F7 Administrator Guide* and *Procedure 8.6, Creating multiple logspaces* in *The syslog-ng Store Box 4 F7 Administrator Guide*.)

Parsing syslog headers adds an extra 18% overhead. You can improve the raw performance of SSB by selecting the **Do not parse** option in the log source. However, in this case you cannot search and filter the host, program, pid, and other fields of these messages.

Disabling flow control on a log source will not throttle back clients, and seems to increase performance. However, it may lead to losing messages.

Factors without significant effect on performance

The following factors have no effect, or only limited effect on the performance of SSB.

- Number of plain TCP connections to the log sources of SSB, up to around 5000 connections.
- Number of SSL/TLS TCP connections to the log sources of SSB, up to around 1000 connections.
- SSB can process about 5% more messages using the IETF-syslog protocol than the legacy BSD-syslog messages.
- Enabling debug logging in SSB has no effect: debug logs are related to tracing web access and related operations.
- The **Trusted**, **Use DNS**, **Use FQDN** settings have limited effect on performance, provided that DNS is correctly set up. Internally there is a DNS cache in syslog-ng. (For details on these settings, see *Procedure 7.3, Creating syslog message sources in SSB* in *The syslog-ng Store Box 4 F7 Administrator Guide*.)

Overall performance

Depending on its exact configuration and the mix of log formats received, the largest SSB appliance can collect and index up to 100,000 messages per second (100k EPS) for sustained periods.

3. Introduction to SSB search algorithms

This section describes how SSB indexes messages and stores data. It gives you insights to understanding and interpreting the search performance of SSB

Data layout per log store

Level 1: File system directories organized per year and day of month (YYYY/MM-DD).

Level 2: Log messages are stored in a single file per day. There is also a file that lists index files (level 3) related to distinct time intervals inside the day.

Level 3: Index file that holds an ordered list of tokens processed when SSB received the logs. For each token there is a list of unique identifiers that points to the messages that contained the token.

Search algorithm overview

Every SSB search must include the time interval we search in. This information is used to find the days, thus the Level 2 and 3 files that need to be searched. Note SSB stores and searches the log messages based on the time SSB received them (the so-called processed timestamp), and not based on the timestamp included in the log message. The reason is that SSB collects logs in real-time, and also the timestamp in the log messages may not be reliable or complete.

Tokens are the words separated by the delimiters set for the logspace (for details, see *Procedure 8.1.2, Configuring the indexer service* in *The syslog-ng Store Box 4 F7 Administrator Guide*).

On Level 3, SSB looks up the tokens that match the basic expressions in the search query. Since the tokens are stored in alphabetic order, this lookup is very fast for exact searches. If the token contains wildcards (`*` or `?`, then potential matches are checked individually.

At this point SSB has the list of message identifiers it needs to calculate AND, OR, NOT expressions and finalize search results per day. Getting the final result simply means repeating the procedure for all the days that are requested in the search interval.

4. Search performance (SSB T4)

Overview

In this section we describe SSB search algorithm performance, measured from starting a search to returning the first 100 results. When a search is executed, SSB calculates the unique identifiers of every search results, without loading the individual messages. The actual messages are loaded temporarily only when requested on the user interface or the RPC API.

This means that it can easily happen that calculating the results takes under a second, but fetching all the resulting messages takes minutes, because it takes time to read the messages from the disk and return them. This also means that the size of the messages has no impact on the memory usage of search.

We have conducted our tests using a real life logspace containing 200 million log entries (about 9.1Gb compressed). We executed the searches directly on SSB to avoid network and caching effects. The test hardware was an SSB T4 appliance, but the response times are very similar on SSB T10 appliances as well. For SSB T1, response times are higher by a factor of x2.5 on the average.

Search for all

The most basic search expression is the empty (or `*`) search, which searches for any message stored in a logspace. This search is immediate with little memory usage, since it only involves looking up the unique identifier ranges intersecting with the time range of the search itself.

Simple search expressions

Search for specific token:

```
Example1: username
Example2: restart
```

The simplest search expression is a specific token, like `login`. Tokens are the words separated by the delimiters set for the logspace (for details, see *Procedure 8.1.2, Configuring the indexer service* in *The syslog-ng Store Box 4 F7 Administrator Guide*).

For 200 million logs, searching for a token takes between 1-5 seconds, and the used memory is roughly the same number of bytes as the number of results.

Wildcards:

```
Example1: user?ame
Example2: system*
Example3: *tool
```

You can specifying part of a token, or add `*`, `?` characters after and/or in front of the token. For example `*pple` or `appl?`.

Search times depend on how many letters are known of the token, especially at the front of the token. The worst case is when the search expression starts with the wildcard, for example `*pple`, which would take between 30-60 seconds to search for in 200 million messages. Searching for `appl*` takes around 9 seconds. The absolute worst case is `*?` where no letter is known, which takes 80 seconds.

Memory consumption is a sum of the number of eventual results plus at most the size of the biggest index file involved in the search. The size of the index file depends on the **Memory limit** setting of the logspace. The higher the limit, the larger the index files. (For details, see *Procedure 8.1.2, Configuring the indexer service* in *The syslog-ng Store Box 4 F7 Administrator Guide*.)

Excluding tokens:

```
Example1: NOT user
Example2: NOT window*
```

You can exclude tokens from a search using the NOT keyword, as in "`NOT apple`, `NOT *pple`, and so on. Such search takes slightly longer than searching for the same expression without NOT. The memory used is roughly the same.

Complex search expressions

Search expressions can be combined with the AND and OR keywords to create complex search expressions. A negation of a complex search expression with NOT keyword remains a complex search expression. Note that SSB automatically optimizes certain search expressions before evaluating them, for example, `expression AND *` becomes `expression`, `NOT NOT expression` becomes `expression`.

Union of searches (OR):

```
Example1: username OR pass*
Example2: server1 OR server32 OR server50 OR server3
```

Response time can be calculated by adding up the response times of the searches included in the OR expression. The actual OR operation is extremely efficient, so there is little additional overhead.

Intersection of searches (AND):

```
Example1: user AND login AND fail
Example2: user AND NOT close*
```

The maximal response time can be calculated by adding up the response times of included searches in the AND expression. The actual AND operation is extremely efficient, so there is little additional overhead.

Statistics on search results:

Same response times and memory consumption expected as for regular searches.

5. Log collection and search

Measurement setup: An indexed logspace was set up with all fields indexed. 25,000,000 log messages were sent via TLS source into the logspace every hour for 24 hours to represent "historical" data. Then we kept sending 25,000,000 log messages every hour, while executing an increasing number of parallel search queries to simulate 1 to 29 concurrent busy users.

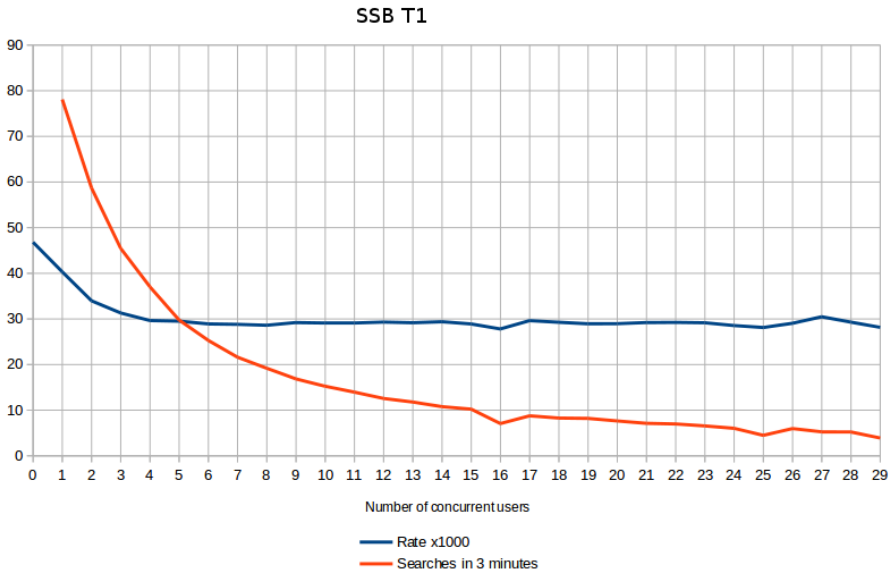
User simulation was achieved by executing the same RPC API queries that a front end (i.e. internet browser) would send. These are: splitting the time interval in question into 30 equal parts and calculating the number of results per interval (these are the bars on the search interface) and also fetching the first 200 results (part of this is show at the bottom of the search interface).

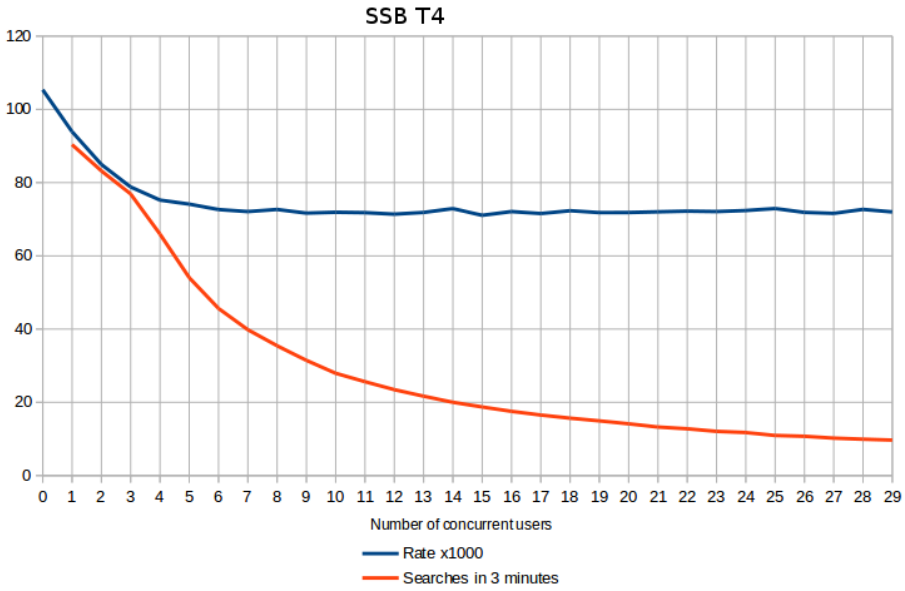
Measurement quantities:

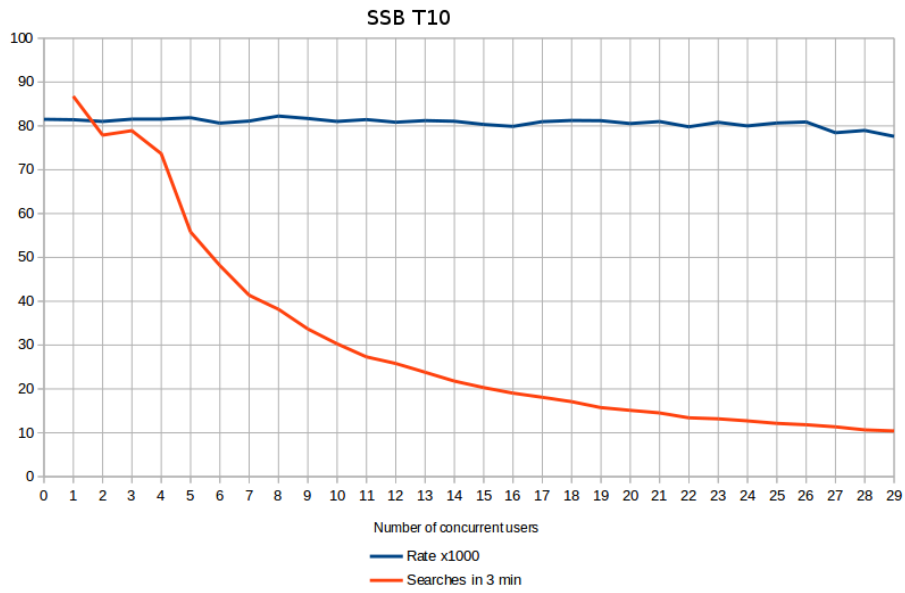
- Rate in 1000x message/second, that is, how many log messages we can send into SSB while running searches at the same time.
- Number of searches finished in 3 minutes of the test.

Case 1

The search queries are very simple exact token searches. Memory limit for the logspace is 1024Mb. Search was for X<n> OR Y<m> with variable n,m numbers to avoid effects of the query cache of SSB.

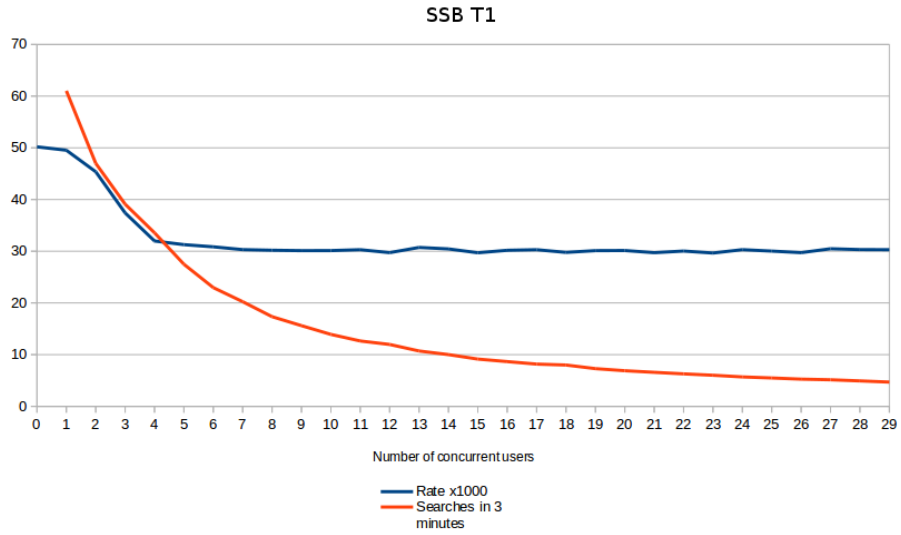






Case 2

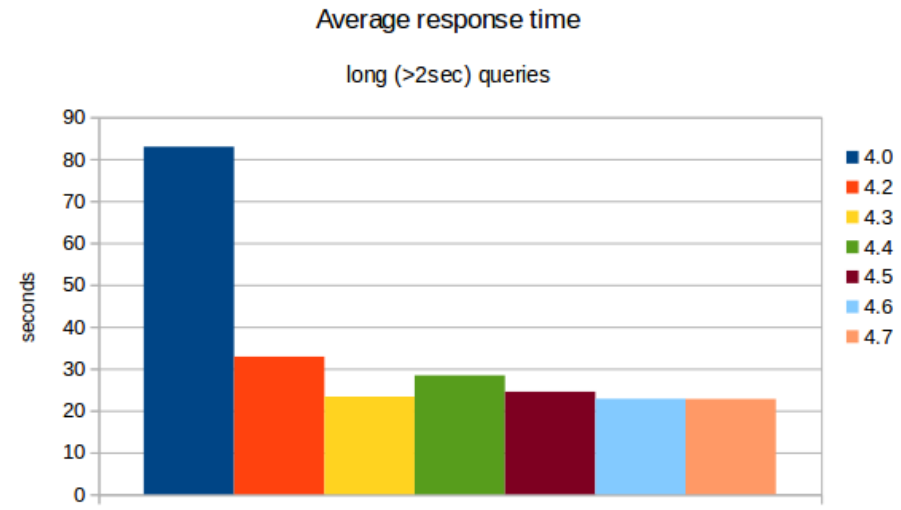
Addition of *Z to the start or the query to force much more disk IO usage. Results were the same regarding throughput and obviously lower for number of searches.

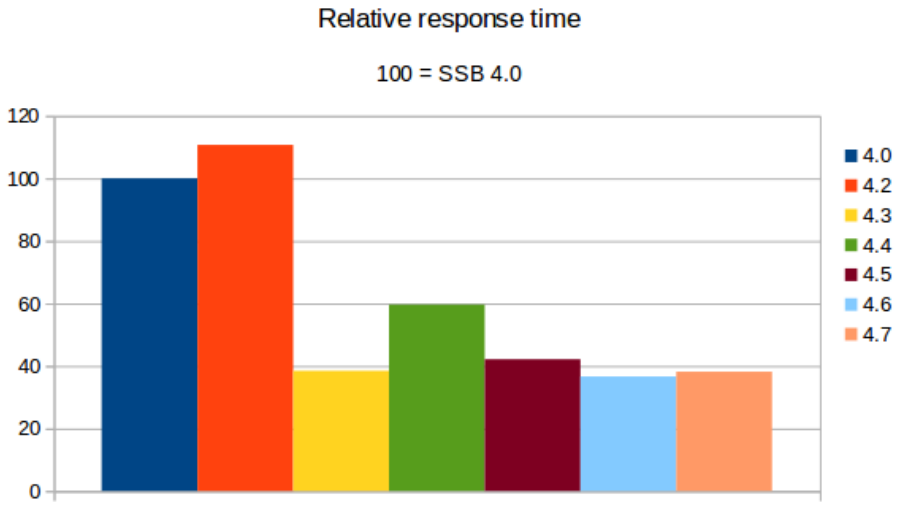

SSB T1
70
60
50
40
30
20
10
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
Number of concurrent users
Rate x1000
Searches in 3 minutes

6. Performance of SSB versions

The following charts show how the search performance of SSB has improved, using SSB 4 LTS (4.0) as a baseline. We used relatively longer queries, where the response time was at least 2 seconds, so that the results are easier to measure and compare. Otherwise, various other factors could have distorted the measurements. Note the following point:

- The structure of index files was optimized in SSB 4 F2, greatly increasing the search performance of SSB. If searching is slow in your SSB and you are using SSB 4 LTS, consider upgrading to a newer version. (Note that this will affect only the index files created after the upgrade.)
- The search algorithms were optimized in SSB 4 F2, decreasing the memory usage of search by an average of 80%. If search causes high memory consumption in your SSB and you are using SSB 4 LTS, consider upgrading to a newer version.





7. Summary

The test measurements show that the processing capabilities and search performance of syslog-ng Store Box have increased significantly since version 4 LTS, and that SSB is capable of receiving and processing high-volume log traffic. The largest SSB appliance is capable of scaling up to 100,000 event per second (100k EPS).

If the search performance of SSB is not adequate in your environment (search is slow, or greatly increases the memory consumption), check the version of your SSB. If you are using SSB 4 LTS, consider upgrading to a newer version.

If you have questions about the performance of SSB, or need help in optimizing the configuration of your SSB appliance, contact our Service Delivery department at <servicedelivery@balabit.com>

7.1. About Balabit

Balabit is an international IT security vendor, founded in Budapest, Hungary. Balabit is a leading provider of contextual security technologies with the mission of preventing data breaches without constraining business. It operates globally through a network of local offices across the United States and Europe together with partners.

Balabit's Contextual Security Intelligence™ strategy protects organizations in real-time from threats posed by the misuse of high-risk and privileged accounts. Solutions include reliable system and application Log Management with context-aware data ingestion, Privileged User Monitoring, and User Behavior Analytics. These technologies can identify unusual user activities, and provide deep visibility into potential threats. Working in conjunction with existing control-based strategies, Balabit enables a flexible and people-centric approach to improve security without adding additional barriers to business practices.

Founded in 2000 and headquartered in Luxembourg, Balabit has a proven track record including over twenty Fortune 100 customers, amongst over 1,000,000 corporate users worldwide. For more information, visit *www.balabit.com*.

To learn more about commercial and open source Balabit products, request an evaluation version, or find a reseller, visit the following links:

- *syslog-ng Store Box (SSB) homepage*
- *Product manuals, guides, and other documentation*
- *Contact us and request an evaluation version*
- *Find a reseller*

All questions, comments or inquiries should be directed to <info@balabit.com> or by post to the following address: Balabit SA 1117 Budapest, Alíz Str. 2 Phone: +36 1 398 6700 Fax: +36 1 208 0875 Web: *https://www.balabit.com/*