Speed up your searches!

Satoshi Kawasaki | Splunk4Good Ninja

September 28th, 2017 | Washington, DC
During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC.

The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

Splunk, Splunk>, Listen to Your Data, The Engine for Machine Data, Splunk Cloud, Splunk Light and SPL are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names, or trademarks belong to their respective owners. © 2017 Splunk Inc. All rights reserved.
Bio: Satoshi Kawasaki
Splunk4Good Ninja

BS in Aerospace Engineering from Georgia Tech

► Also joined Splunk in 2013
  • 3 years of Professional Services (PS)
  • 1+ year of Splunk4Good

► Unofficially became a dashboard/visualization specialist in PS
  • .conf 2014: I Want that Cool Viz in Splunk!
  • .conf 2015: Enhancing Dashboards with JavaScript!

► Doing 3 talks this year
  • .conf 2017: Speed up your searches!
  • .conf 2017: Splunking to fight human trafficking
  • .conf 2017: Splunking the 2016 presidential election
Splunk4Good

Big data can make a big difference

- $100 million Splunk Pledge has issued licenses and training worth over $6 million
- Provide workforce training to veterans and opportunity youth to train the workforce of tomorrow
- Engaging our partners in initiatives to promote STEM and develop shared solutions for humanitarian response and human trafficking
- Supporting life-changing research at top universities
- More than 70,000 hours of paid volunteer time
Dashboards are like web pages
Because all good searches become dashboards

"For every one second delay, conversions dropped by 7%"

"2 seconds is the threshold for ecommerce website acceptability. We aim for under a half second."

"For every one second past 2 seconds a Splunk dashboard loads, the user becomes 20% more likely to open YouTube, Facebook, or 4chan."
How does acceleration work?

Nothing in this world is free

Increase speed at the cost of space!

Luckily, disk space is much cheaper than processors!
Table of contents
Also known as the "summary" or .tsidx

▶ Scheduled searches[1]
▶ Post-process searches[1]
▶ Event sampling
▶ Summary indexing
▶ Report acceleration
▶ DATA MODEL ACCELERATION
▶ Batch mode search parallelization[2]

[1] For dashboards
[2] This is actually an indexer setting
The baseline search
Cisco Meraki providing free wifi in refugee camps around Greece

A sample of 2,251,967 raw events from July 19th, 2017

The baseline search takes **77s**:

```
index=meraki sourcetype=meraki_syslog
log_type=urls
| stats dc(mac)
```
Scheduled searches

"It's my search and I need it now!"
Scheduled search
For dashboard panels

Panel status shows the 39 minute "delay" in the scheduled search.

Job Inspector (or "View Recent" from "Searches, reports, and alerts") shows how long the search actually took and when the search last ran.
Scheduled search
Pros and cons

▶ Searches instantly load from disk
▶ Good for "static" dashboards (like single value KPIs for TV displays)
▶ Better than saving to lookups for static data

▶ Less flexibility on search parameters, like you can't increase the time range
▶ Results delayed up to the scheduled interval
▶ Managing a saved search per panel could be a pain

[1]Unless you're really working with unreliable test data
Post-process searches

One construction worker working, the rest standing
Post-process searches
For dashboards

Two searches driven by one base search (aka the "data cube").

Both post-process searches will complete at the same time.
Post-process search
Pros and cons

- Post-process searches share the same processing usage of the base search.
- As long as the base search doesn't change, changes in post-process is very fast (ie using $tokens$).
- Less validation on search results when post-processing from a "data cube".

- Must be done in Simple XML (no UI option as of Splunk 6.6).
Event sampling

"We're gonna need a bigger sample"
Event sampling

Sampling 1:10

No sampling covers 2,251,967 events (baseline)

1:10 sampling covers 225,672 events

Generally,
1:10 is 10× faster,
1:100 is 100× faster, etc.
Event sampling

Pros and cons

- Easiest way to speed up a search
- No prerequisites to use event sampling
- Good for ratios (ie pie charts)

- Results are approximates with inherent sampling errors
- A big assumption is that the data is uniform enough
- Certain statistical functions are almost useless in sampling (like total count, sum, dc, etc.)
Summary indexing

Summary indexing (SI)
Searching against the summary index

- Original search:
  ```bash
  index=meraki sourcetype=meraki_syslog log_type=urls | stats dc(mac)
  ```

- Summary index search:
  ```bash
  index=summary search_name=conf_2017_si | stats dc(mac)
  ```
Summary indexing (SI)

The summarizing search that goes into the SI

Summary-populating search called "conf_2017_si" runs every hour and looks back one hour[1]:

```bash
index=meraki sourcetype=meraki_syslog log_type=urls | sistats dc(mac) by device
```

07/19/2017 06:00:00 -0700, search_name=conf_2017_si,
search_now=1500519600.000, info_min_time=1500516000.000,
info_max_time=1500519600.000,
info_search_time=1501727194.366, device=GRE_040_AP5,
psrsvd_ct_mac=408, psrsvd_gc=408, psrsvd_v=1,

"Mysterious" fields created by sistats

[1]Backfilled the SI using:

```
./splunk cmd python fill_summary_index.py -app conf_2017 -name conf_2017_si -et 1500447600 -lt 1500534000 -owner admin
```
Summary indexing
How is SI fast?

Original index with 2,251,967 events (baseline)
SI with 494 events
Summary indexing

Pros and cons

> Also useful for having a "cleaner" copy of the data or hardcoding calculated or lookup values to the summary

> Has all the same functionalities of an index: RBAC, data retention, clustering replication, etc.

> Can't go more granular than the summary's scheduled interval

> Can have gaps or overlaps

> Backfilling is a manual python script

> Impossible to search outside the summarized time range

> Messing up the summary is the worst
Report acceleration

The "that was easy" button
Report acceleration (RA)
Simply check a box and select a summary range

Create a saved search and enable RA

Some similar searches (even ad-hoc) will automagically use the RA summary
Report acceleration (RA)

Pros and cons

- Very easy to enable
- Has a summary time range to easily control the size of the RA
- Searching outside the summary time range will automatically fall back to a regular search
- Similar searches automagically uses the RA summary

- Similar searches automagically not using the RA summary (just switching the order of the search terms tricks Splunk to not use the RA summary, i.e., `foo=A bar=B` vs `bar=B foo=A`
DATA MODEL ACCELERATION!
The big daddy of search acceleration
DATA MODEL (DM) ACCELERATION
Regular vs tstats search format

- Regular search:
  ```
  index=meraki sourcetype=meraki_syslog log_type=urls
  | sistats dc(mac) by device
  | stats dc(mac)
  ```

- DM (tstats) search:
  ```
  | tstats|prestats=t dc(all.mac) from datamodel=conf_2017
  by all.device
  | stats dc(all.mac)
  ```
DATA MODEL (DM) ACCELERATION

Regular vs tstats search format

Simple example:

```
index=meraki sourcetype=meraki_syslog log_type=urls | stats dc(mac)
```

Advanced example:

```
index=meraki sourcetype=meraki_syslog log_type=urls | sistats dc(mac) by device | stats dc(mac)
```

```
| tstats prestats=t dc(all.mac) from datamodel=conf_2017 by all.device | stats dc(all.mac)
```
DATA MODEL (DM) ACCELERATION

Creating the data model

Before using tstats, you must create a DM[1]

Keep this name short! (you'll be typing this a lot)

Only root events can be accelerated

List the fields you will use later in tstats

[1]You can actually use tstats without a DM, but you can only use index-time fields (default fields like host, sourcetype, etc. or indexed extraction fields)
You can actually use `tstats` searches on an unaccelerated DM.

This way you can review and check that all fields are accounted for before accelerating the DM.

If a `tstats` searches outside the summary range, then it will automagically convert that part to a regular search (like RA).
DM acceleration basically creates a compressed, optimized summary table (.tsidx files) on the indexers where

- rows = # of root events within the summary range
- columns = # of fields in the DM

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>_time</td>
<td>host</td>
<td>...</td>
<td>device</td>
<td>mac</td>
</tr>
<tr>
<td>1501634605</td>
<td>meraki</td>
<td>...</td>
<td>GRE_003_AP2</td>
<td>00:00:3F:2E:4B:3A</td>
</tr>
<tr>
<td>1501634662</td>
<td>meraki</td>
<td>...</td>
<td>GRE_003_AP2</td>
<td>00:03:AB:11:4B:7D</td>
</tr>
<tr>
<td>1501634705</td>
<td>meraki</td>
<td>...</td>
<td>GRE_003_AP3</td>
<td>00:08:22:72:6C:3A</td>
</tr>
</tbody>
</table>

Therefore size of DM ~ rows × columns
DATA MODEL (DM) ACCELERATION

DM acceleration cost

DM summary lives on the indexers[1] and is only 37 MB total!

Is this worth speeding up the search by almost 100×?

YES!

[1]DM summary lives in
SPLUNK_DB/<index_name>/datamodel_summary/<bucket_id>_<indexer_guid>/
<search_head_guid>/DM_<app>_<data_model_name>
DATA MODEL (DM) ACCELERATION

Pros and cons

- Reusability: one DM can feed many searches
- Summaries can be replicated in a cluster (not by default)
- Also useful for hardcoding calculated or lookup values to the summary (like in SI)
- Tstats can still search outside the summary range

- Requires creating an accelerated DM first
- May need to manually convert old searches to tstats and not all searches can be converted
- Need to stop and re-accelerate the DM to modify it
- Tstats is only fast for "reducing" searches
Batch mode search parallelization

Because two is better than one
Batch mode search parallelization allows launching multiple search pipelines per qualifying search[1], which are processed concurrently.

Set limits.conf on indexers:

```
[search]
batch_search_max_pipeline = 2
```

- The default is 1
- 2 is the best value (higher values succumbs to diminishing returns)

[1] Only for "batch mode" searches, which are searches that are distributed (i.e., not time-ordered searches like streamstats, transaction, head, etc.)
Batch mode search parallelization

Pros and cons

- Faster searches by using up more resources (IO, processing, and memory)
- Only for the rich
- Only works on "batch mode" searches
Review
The final countdown!

<table>
<thead>
<tr>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scheduled search</strong></td>
</tr>
<tr>
<td>Caching fixed time range search results</td>
</tr>
<tr>
<td><strong>Post-process searches</strong></td>
</tr>
<tr>
<td>Creating a &quot;data cube&quot; to power multiple other searches</td>
</tr>
<tr>
<td><strong>Event sampling</strong></td>
</tr>
<tr>
<td>Randomly sampling every 1 out of X events</td>
</tr>
<tr>
<td><strong>Summary indexing</strong></td>
</tr>
<tr>
<td>Reducing the number of events by reducing the time &quot;resolution&quot; to a new index</td>
</tr>
<tr>
<td><strong>Report acceleration</strong></td>
</tr>
<tr>
<td>The lazy version of data model acceleration</td>
</tr>
<tr>
<td><strong>DATA MODEL ACCELERATION</strong></td>
</tr>
<tr>
<td>Create a data model, then use it via tstats</td>
</tr>
<tr>
<td><strong>Batch mode search acceleration</strong></td>
</tr>
<tr>
<td>Don't worry about this unless your Splunk is heavily underutilized.</td>
</tr>
</tbody>
</table>
Mix and match!

"No seriously, I have nothing to wear!"
Mix and match!
The sky is the limit

Examples:
- DMs off of SI
- Post-process searches off of a scheduled search
- RA off of SI
- Tstats to create SI
- Scheduled search off of tstats
Closing remark

Satoshi Kawasaki | Splunk4Good Ninja
Thank You!

Shout-out to Eric Merkel, my content delivery manager!
And to all of my fellow PSers and awesome former clients!

Don't forget to rate this session in the .conf2017 mobile app