Speed up your search!

Satoshi Kawasaki
Splunk for Good Ninja

splunk> .conf19

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or plans 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 may differ materially. The forward-looking statements made in the this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, it may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements made herein.

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 functionalities described or to include any such feature or functionality in a future release.

Splunk, Splunk>, Turn Data Into Doing, 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. © 2019 Splunk Inc. All rights reserved.

Bio: Satoshi Kawasaki

BS in Aerospace Engineering from Georgia Tech

hobbes3

1. Joined Splunk in 2013
   - 3 years in Splunk Professional Services (PS)
   - 3+ years in Splunk for Good
2. Previous conf talks:
   - conf14: I want that cool viz in Splunk!
   - conf15: Enhancing dashboards with javascript!
   - conf17: Speed up your searches!
   - conf17: Splunking to fight human trafficking!
   - conf17: Splunking the 2016 presidential election!
3. This year's conf talks:
   - conf19: Speed up your searches! YOU ARE HERE
   - conf19: Splunking refugees with help from NetHope and Cisco!
   - conf19: Splunking the 2018 midterm election!

splunk> .conf19

Splunk for Good

Big data can make a big difference

$100 million Splunk Pledge has issued licenses and training worth over $40 million.

WOUNDED WARRIOR PROJECT

Provide workforce training to veterans and opportunity youth to train the workforce of tomorrow.

NETHOPE

Engaging our partners in initiatives to promote STEM and develop shared solutions for humanitarian response and human trafficking.

TEAM RUBICON

Supporting life-changing research at top universities.

npower

More than 100k hours of paid volunteer time.

Dashboards are like web pages

Because all good searches become dashboards

amazon

"For every one second [website] delay, conversions dropped by 7%.”

Google

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

"For every one second delay of a Splunk dashboard, the user becomes 7% more likely to go view YouTube, Facebook, or Reddit instead."

How does acceleration work?

Nothing in this world is free

Increase speed at the cost of space![1]

Luckily, disk space is much cheaper than processors!

[1] Another way to look at it is sacrificing search-time flexibilities (like schema-on-the-fly field extractions) to gain speed.

Table of Contents

Also know as the .tsidx

- Scheduled searches
- Post-process searches
- Event sampling
- Summary indexing
- Report acceleration
- **DATA MODEL ACCELERATION**
- Metrics
- Batch mode search parallelization

The baseline search

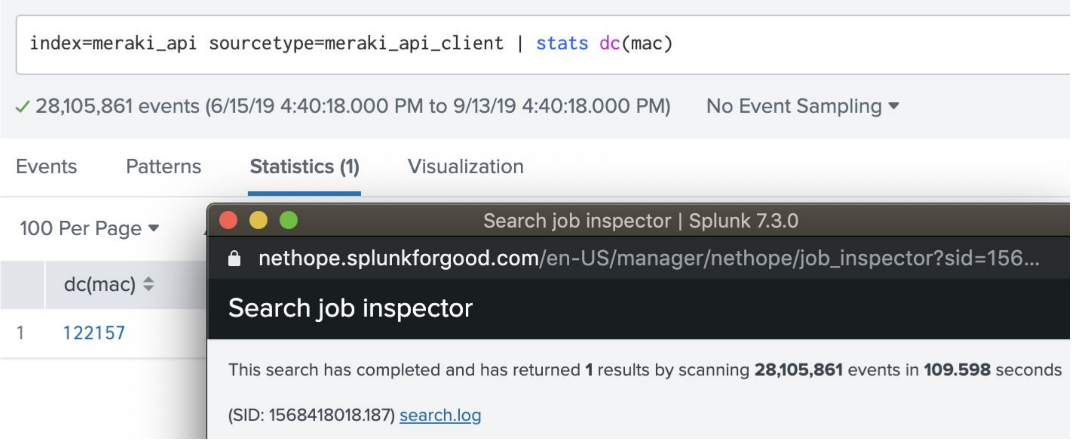
Cisco Meraki providing free wifi in NetHope refugee camps

109s

28 million raw events from the last 90 days.

The baseline search takes 109 seconds:

```
index=meraki_api sourcetype=meraki_api_client
| stats dc(mac)
```



```
index=meraki_api sourcetype=meraki_api_client | stats dc(mac)
```

✓ 28,105,861 events (6/15/19 4:40:18.000 PM to 9/13/19 4:40:18.000 PM) No Event Sampling ▾

Events
Patterns
Statistics (1)
Visualization
100 Per Page ▾

	dc(mac)
1	122157

Search job inspector | Splunk 7.3.0
nethope.splunkforgood.com/en-US/manager/nethope/job_inspector?sid=156...

Search job inspector

This search has completed and has returned **1** results by scanning **28,105,861** events in **109.598** seconds

(SID: 1568418018.187) search.log

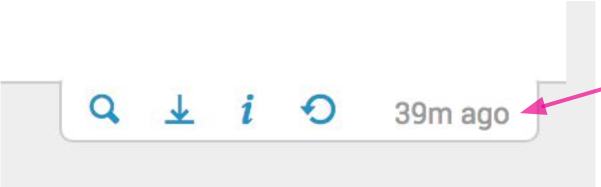
Scheduled searches

"It's my search and I need it now!"

Scheduled search

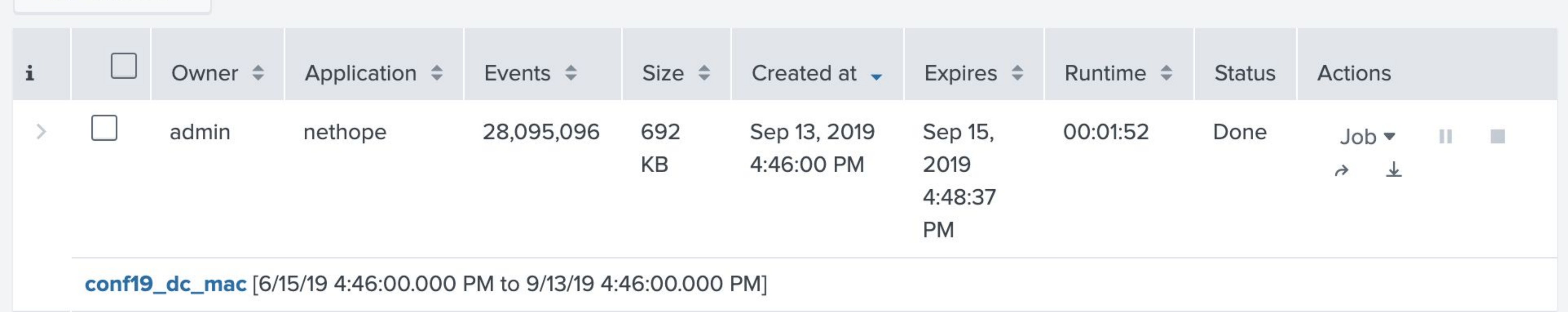
For dashboard panels

Easiest way is to "Edit Search" > "Convert to Report".



Panel status shows the result is 39 minute "old" in the scheduled search.

<1s



i	□	Owner	Application	Events	Size	Created at	Expires	Runtime	Status	Actions
>	□	admin	nethope	28,095,096	692 KB	Sep 13, 2019 4:46:00 PM	Sep 15, 2019 4:48:37 PM	00:01:52	Done	Job ▾

conf19_dc_mac [6/15/19 4:46:00.000 PM to 9/13/19 4:46:00.000 PM]

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[1].

👎

- You can't change the time range.
- Also can't use `$tokens$`.
- Results delayed up to the scheduled interval.
- Managing a saved search for many panels is annoying.

[1]Unless you're really working with test data and you don't care a large lookup potentially causing a large replication bundle (can be blacklisted via `distsearch.conf`).

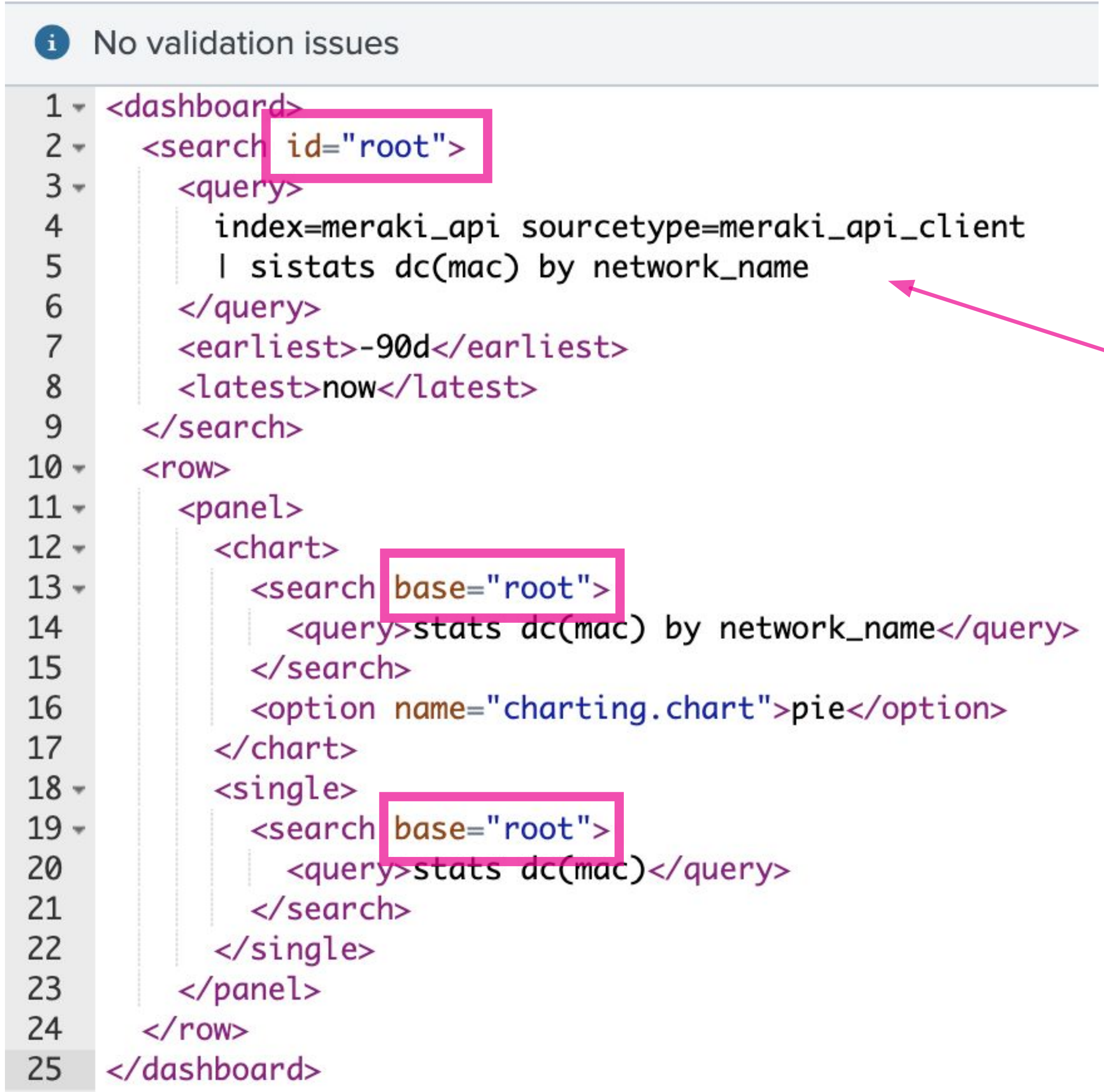
Post-process searches

It's a "team" project

Post-process searches

For dashboards

N/A



No validation issues

```
<dashboard>
  <search id="root">
    <query>
      index=meraki_api sourcetype=meraki_api_client
      | sistats dc(mac) by network_name
    </query>
    <earliest>-90d</earliest>
    <latest>now</latest>
  </search>
  <row>
    <panel>
      <chart>
        <search base="root">
          <query>stats dc(mac) by network_name</query>
        </search>
        <option name="charting.chart">pie</option>
      </chart>
      <single>
        <search base="root">
          <query>stats dc(mac)</query>
        </search>
      </single>
    </panel>
  </row>
</dashboard>
```

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

Both post-process searches will basically complete at the same time.

splunk> .conf19

Post-process search

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.).

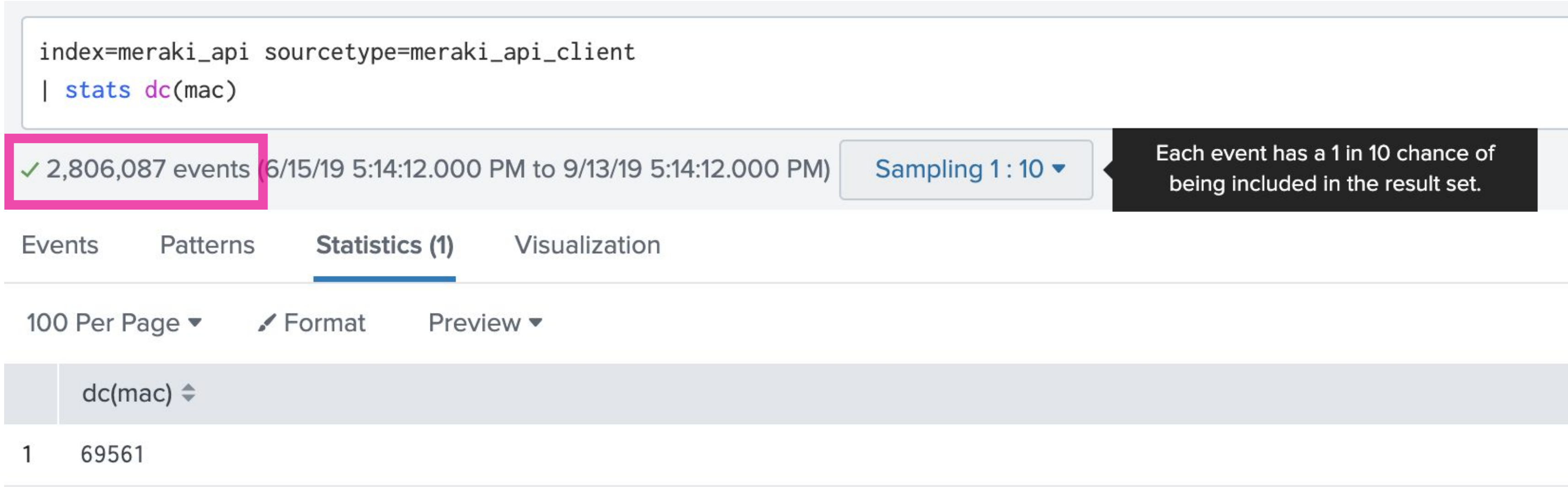
Summary indexing

Search. Reduce. Recycle.

Event sampling

Sampling 1:10

20s



```
index=meraki_api sourcetype=meraki_api_client
| stats dc(mac)
```

✓ 2,806,087 events (6/15/19 5:14:12.000 PM to 9/13/19 5:14:12.000 PM) Sampling 1 : 10 ▾

Each event has a 1 in 10 chance of being included in the result set.

	dc(mac)
1	69561

- No sampling covers 28 million events (baseline).
- 1:10 sampling covers 2.8 million 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

Search. Reduce. Recycle.

Summary indexing (SI)

Searching against the summary index

<1s

- Original search:

```
index=meraki_api sourcetype=meraki_api_client
| stats dc(mac)
```

- Summary index search:

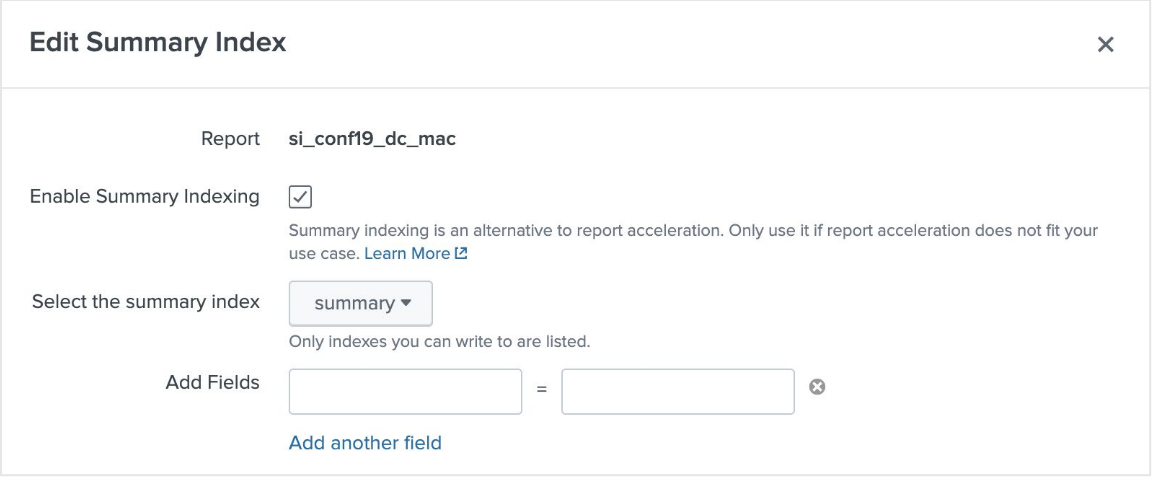
```
index=summary search_name=si_conf19
| stats dc(mac)
```

Summary indexing (SI)

The summarizing search that goes into the SI

Summary-populating search called "si_conf19" runs every day and looks back one day[1]:

```
index=meraki_api sourcetype=meraki_api_client
| sistats dc(mac) by device
```



Edit Summary Index ✕

Report si_conf19_dc_mac

Enable Summary Indexing ☑

Summary indexing is an alternative to report acceleration. Only use it if report acceleration does not fit your use case. Learn More

Select the summary index: summary ▾

Only indexes you can write to are listed.

Add Fields [ ] = [ ]

Add another field

```
09/13/2019 16:22:00 -0700, search_name=si_conf19_dc_mac,
search_now=1568420520.000, info_min_time=1568416920.000,
info_max_time=1568420520.000,
info_search_time=1568420520.393, psrsvd_v=1,
psrsvd_gc=1233, psrsvd_ct_mac=1233,
psrsvd_vm_mac="#1::+8Z1jQN0zYv6/ILdexfhCv4jZrdMtrlNx/+lLx
iLQgQ#;1;#1::+Awh2xqIa0eb9QVrXux0kUDRcMYyxqAekHA8JxfoPWk#
;1;#1::+B/bF4p9xFrTLTcKPEPQolqjmMRdCjG1UFS8ugkjcj0#;1;#1:
:+BNZDdemrCEOuRvqtJdLE3BnimmWWKweqcKMY0PMnTQ#;1;#1::+Cdtu
lHdwBbl/A0UPyYH58Koo9+BFHEI22G4jLKq6TE#;1; "
```

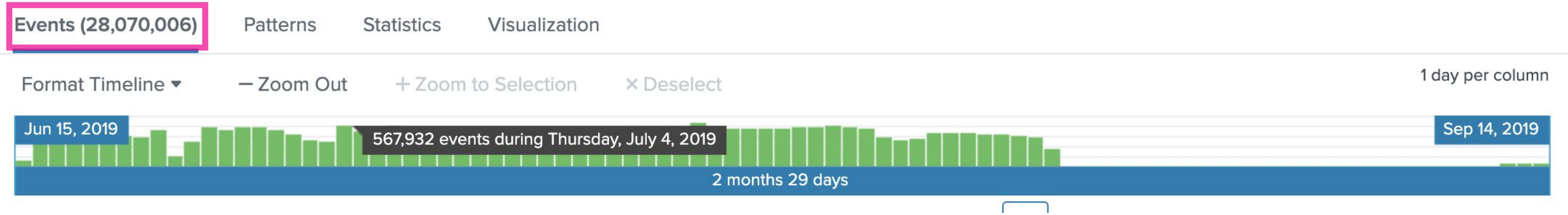
"Mysterious" `psrsvd_` fields created by sistats

[1]Backfilled the SI using:

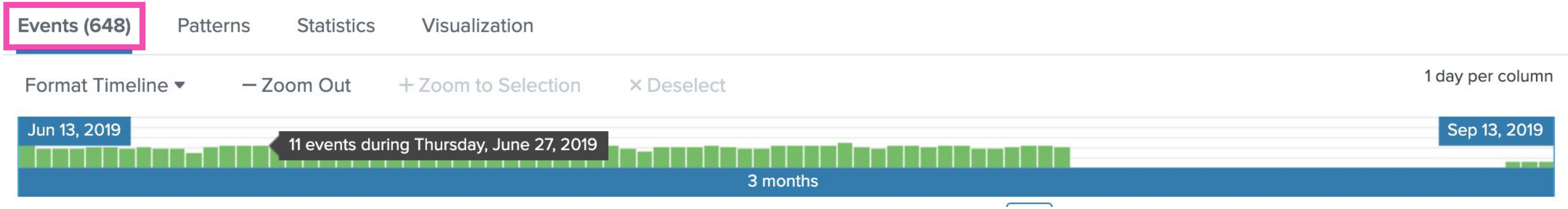
```
./splunk cmd python fill_summary_index.py -app conf19 -name si_conf19_dc_mac -et 1560472279 -lt 1568421079 -owner admin
```

Summary indexing

How is SI fast?



567,932 events during Thursday, July 4, 2019



11 events during Thursday, June 27, 2019

- Original index with 28 million events (baseline).
- SI with 648 events.

Summary indexing

Pros and cons

- Can significantly reduce the number of events to search.
- 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[1].
- Impossible to search outside the summarized time range.
- Messing up the summary is painful to fix

Report acceleration

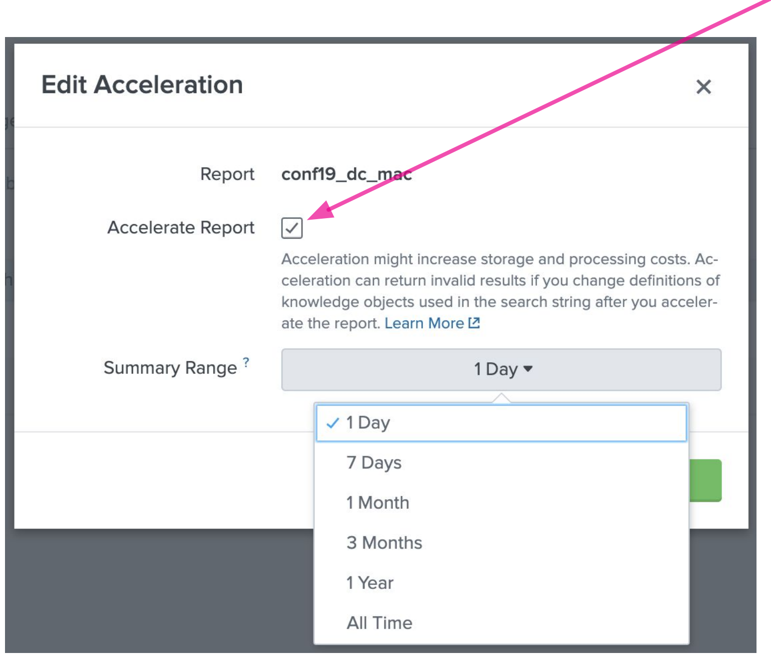
The "that was easy" button

splunk> .conf19

Report acceleration (RA)

Simply check a box and select a summary range

<1s



Edit Acceleration ×

Report conf19_dc_mac

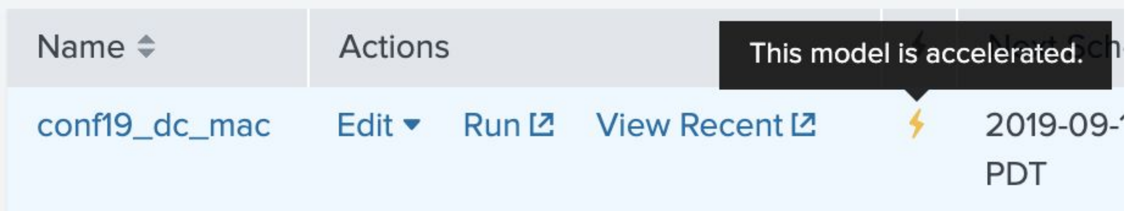
Accelerate Report ☑

Acceleration might increase storage and processing costs. Acceleration can return invalid results if you change definitions of knowledge objects used in the search string after you accelerate the report. Learn More

Summary Range ? 1 Day ▾

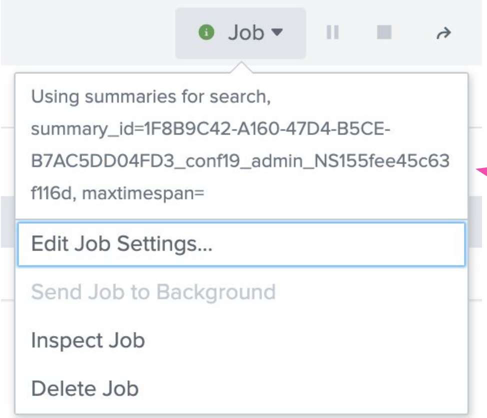
- ✓ 1 Day
- 7 Days
- 1 Month
- 3 Months
- 1 Year
- All Time

Create a saved search and check a box to enable RA



Name ⬍	Actions				Next Sch
conf19_dc_mac	Edit ▾	Run	View Recent	⚡	2019-09- PDT

This model is accelerated.



Job ▾

Using summaries for search, summary_id=1F8B9C42-A160-47D4-B5CE-B7AC5DD04FD3_conf19_admin_NS155fee45c63f116d, maxtimespan=

- Edit Job Settings...
- Send Job to Background
- Inspect Job
- Delete Job

Similar searches (even ad-hoc) will automagically use the RA summary

Report acceleration (RA)

Pros and cons

- Similar searches automagically uses the RA summary.
- 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 *not* use the RA summary (just switching the order of the search terms tricks Splunk to not use the RA summary, ie `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

<1s

- Regular search:

```
index=meraki_api sourcetype=meraki_api_client
| stats dc(mac)
```

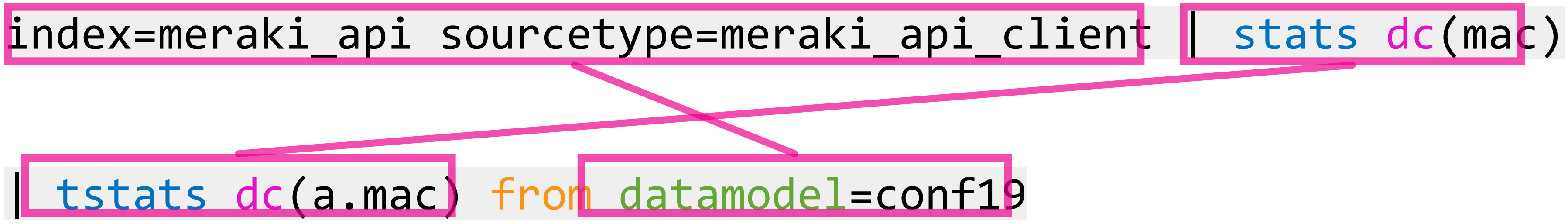
- DM (`tstats`) search:

```
| tstats dc(a.mac) from datamodel=conf19
```

DATA MODEL (DM) ACCELERATION

Regular vs tstats search format

Simple example:



```
index=meraki_api sourcetype=meraki_api_client | stats dc(mac)
```

```
| tstats dc(a.mac) from datamodel=conf19
```

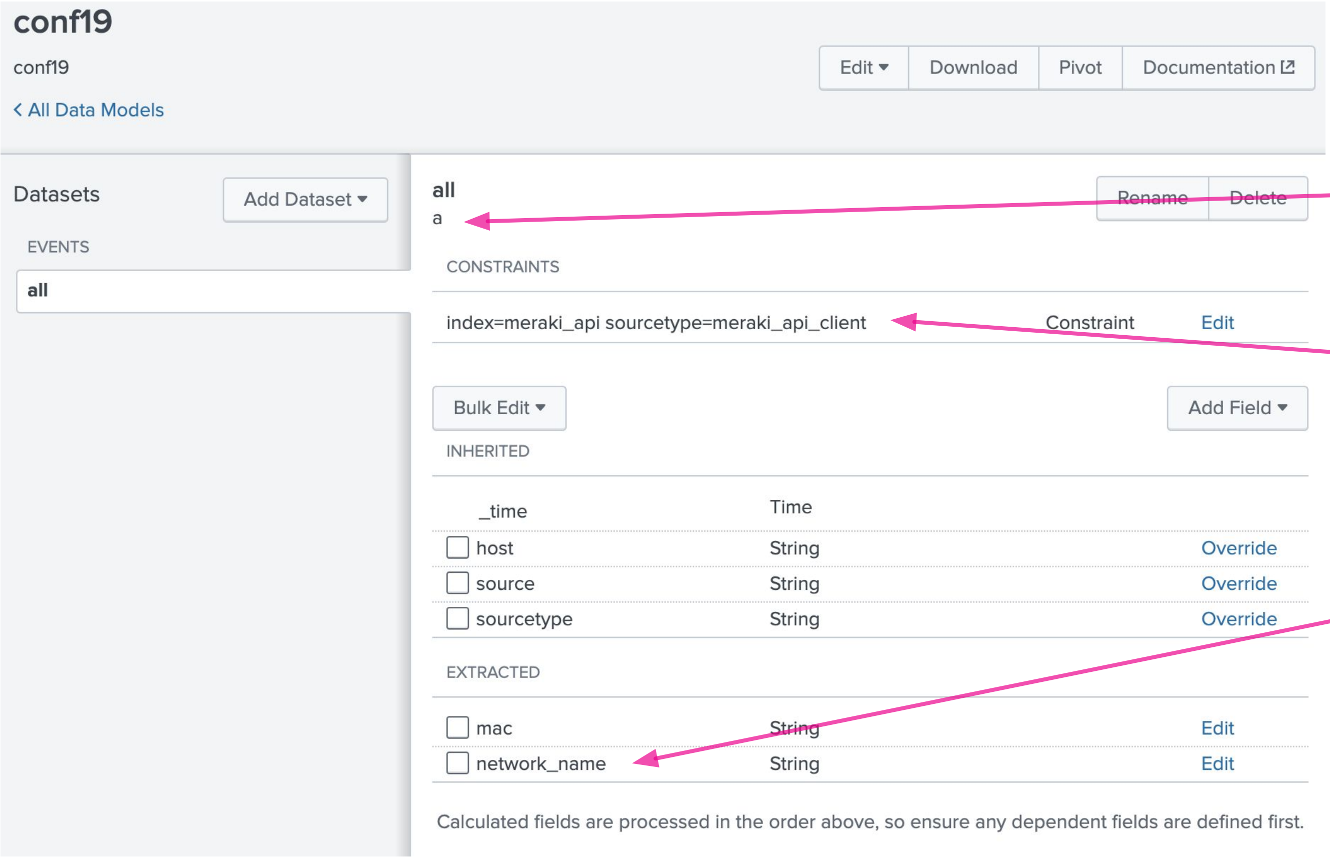
Advanced example:

```
index=meraki sourcetype=meraki_api_client
| timechart dc(mac) by network_name
```

```
| tstats prestats=t dc(a.mac) from datamodel=conf19 by a.network_name _time
| timechart dc(a.mac) by a.network_name
```

DATA MODEL (DM) ACCELERATION

Creating the data model



conf19

conf19

< All Data Models

all

a

CONSTRAINTS

index=meraki_api sourcetype=meraki_api_client	Constraint	Edit

INHERITED

_time	Time	
□ host	String	Override
□ source	String	Override
□ sourcetype	String	Override

EXTRACTED

□ mac	String	Edit
□ network_name	String	Edit

Calculated fields are processed in the order above, so ensure any dependent fields are defined first.

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

Keep this name short like one letter since you'll be typing this a lot!

Only one root event can be accelerated (no pipes or other commands allowed)

List the fields you will use later in tstats

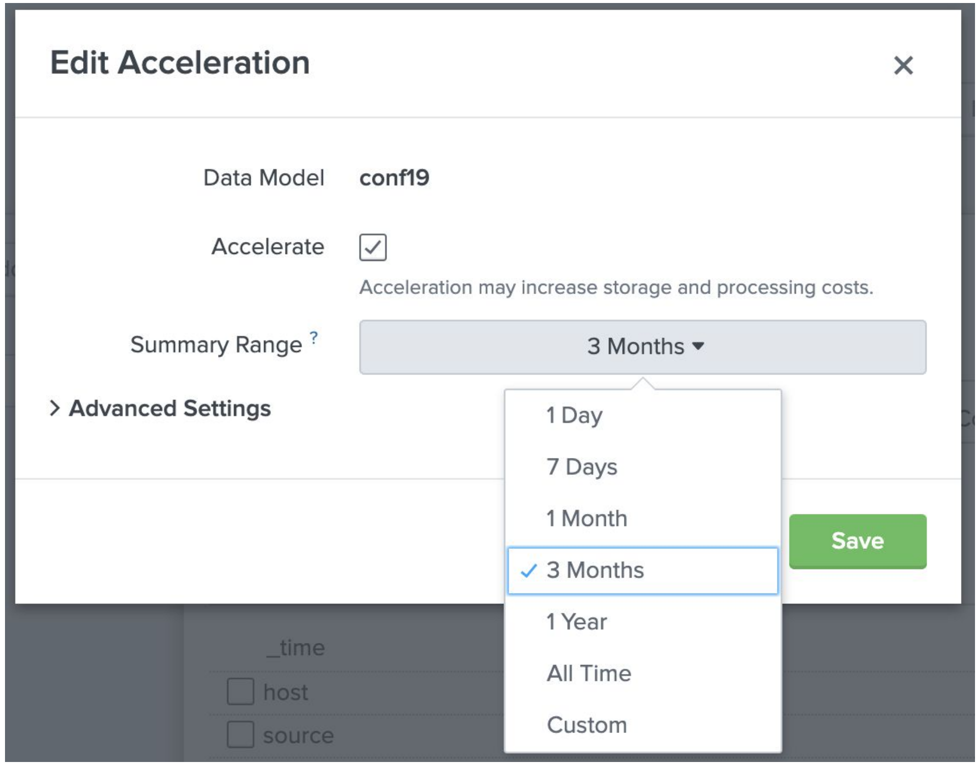
[1]You can actually use tstats without a DM, but you can only use index-time fields (default fields like host, sourcetype, etc. or indexed extraction fields)

DATA MODEL (DM) ACCELERATION

Accelerating the data model

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.



Edit Acceleration

Data Model conf19

Accelerate ☑

Acceleration may increase storage and processing costs.

Summary Range: 3 Months

If a tstats searches outside the summary range, then it will automagically convert that part to a regular search (like RA).

DATA MODEL (DM) ACCELERATION

What really happens when you accelerate a DM

DM acceleration basically creates a compressed, optimized summary table (.tsidx files) on the indexers where

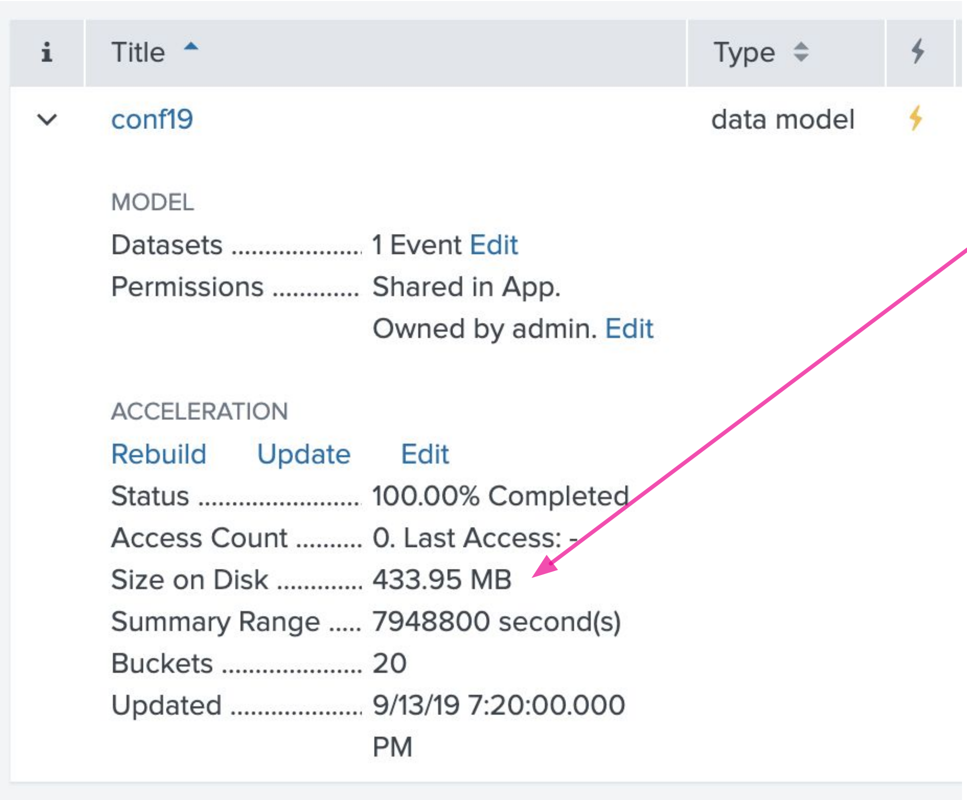
- rows = # of root events within the summary range
- columns = # of fields in the DM

	_time	host	...	network_name	mac
event 1	1501634605	meraki	...	GR-001 Alexandria Ref	00:00:3F:2E:4B:3A
event 2	1501634662	meraki	...	GR-012 Leros-Lepida	00:03:AB:11:4B:7D
event 3	1501634705	meraki	...	GR-023 Ritsona	00:08:22:72:6C:3A
...	...	...	...	...	...

Therefore size of DM ~ rows × columns

DATA MODEL (DM) ACCELERATION

DM acceleration cost



i	Title	Type	⚡
⌄	conf19	data model	⚡

MODEL

Datasets .... 1 Event Edit
Permissions .... Shared in App.
Owned by admin. Edit

ACCELERATION

Rebuild Update Edit

Status .... 100.00% Completed
Access Count .... 0. Last Access: -
Size on Disk .... 433.95 MB
Summary Range ..... 7948800 second(s)
Buckets 20
Updated .... 9/13/19 7:20:00.000 PM

DM summary lives on the indexers[1] and is only 433 MB total for 28 million events!

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.

Metrics

Take the meh out of metrics

Metrics

One number, one asset per event

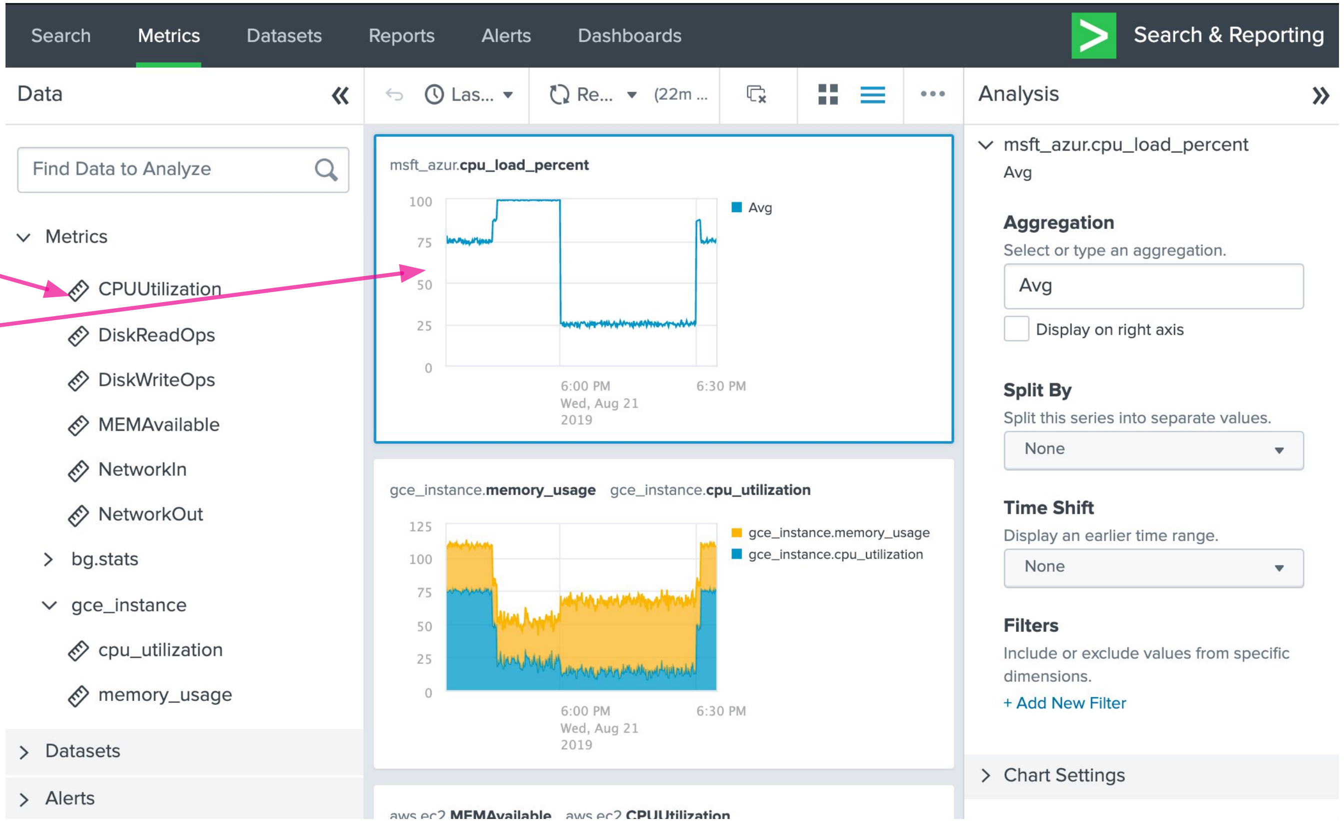
N/A

Metric asset

Metric values

Originally designed for industrial sensors with hierarchical properties

Can be around 20x faster than tstats.



msft_azur.cpu_load_percent
Avg

Aggregation
Select or type an aggregation.
Avg
□ Display on right axis

Split By
Split this series into separate values.
None

Time Shift
Display an earlier time range.
None

Filters
Include or exclude values from specific dimensions.
+ Add New Filter

splunk> .conf19

Metrics

Pros and cons

👍

- Simplistic format.
- Very very fast due to small indexed overhead (no lexicon).
- Has a great UI called Analytics Workspace (formerly known as Metrics Workspace).

👎

- Simplistic format (only one number per event).
- Very specific use cases.
- Metrics is typically tailored around StatsD, collectd, or custom scripts.
- Metrics only works on floating point numbers (no categories).

Batch mode search parallelization

Because two is better than one

Batch mode search parallelization

What it is and where to set this setting

N/A

Batch mode search parallelization allows launching multiple search pipelines per qualifying search[1], which are processed concurrently.

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

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)

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!

Strategy	Time	Short definition
Original baseline search	109s	Good ol' regular search; is slow but has the search-time flexibilities
Scheduled search	<1s	Caching results of a fixed time range search
Post-process searches	N/A	Creating a "data cube" to power multiple other searches
Event sampling	20s	Randomly sampling every 1 out of X events
Summary indexing	<1s	Reducing the number of events by reducing the time "resolution" to a new index
Report acceleration	<1s	The lazy version of data model acceleration
DATA MODEL ACCELERATION	<1s	Create an accelerated data model (a "table"), then search it via `tstats`
Metrics	N/A	A special event format for numerical values of names.
Batch mode search acceleration	N/A	Don't worry about this unless your indexers are heavily underutilized.

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 DM
- Post-process searches off of scheduled search
- RA off of SI
- Tstats to create SI
- Scheduled search off of tstats

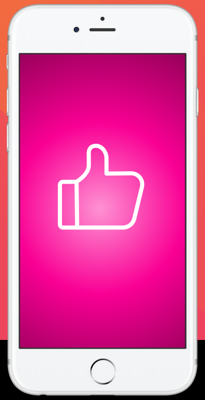
Closing remark

Satoshi Kawasaki | Splunk for Good Ninja

.conf19

splunk>

Thank You !



Go to the .conf19 mobile app to

RATE THIS SESSION

Q&A

Satoshi Kawasaki | Splunk for Good Ninja