Forward-Looking Statements

This presentation may contain forward-looking statements regarding future events, plans or the expected financial performance of our company, including our expectations regarding our products, technology, strategy, customers, markets, acquisitions and investments. These statements reflect management's current expectations, estimates and assumptions based on the information currently available to us. These forward-looking statements are not guarantees of future performance and involve significant risks, uncertainties and other factors that may cause our actual results, performance or achievements to be materially different from results, performance or achievements expressed or implied by the forward-looking statements contained in this presentation.

For additional information about factors that could cause actual results to differ materially from those described in the forward-looking statements made in this presentation, please refer to our periodic reports and other filings with the SEC, including the risk factors identified in our most recent quarterly reports on Form 10-Q and annual reports on Form 10-K, copies of which may be obtained by visiting the Splunk Investor Relations website at www.investors.splunk.com or the SEC's website at www.sec.gov. The forward-looking statements made in this presentation are made as of the time and date of this presentation. If reviewed after the initial presentation, even if made available by us, on our website or otherwise, it may not contain current or accurate information. We disclaim any obligation to update or revise any forward-looking statement based on new information, future events or otherwise, except as required by applicable law.

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. We undertake no obligation either to develop the features or functionalities described, in beta or in preview (used interchangeably), or to include any such feature or functionality in a future release.

Splunk, Splunk> and Turn Data Into Doing 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. © 2023 Splunk Inc. All rights reserved.

splunk> .conf23

© 2023 SPLUNK INC.

I Am Speed!

Searching on Your Own TERMs With Simple Techniques That 99% Aren't Using!

PLA1258C

Jerrod Anderson

Cyber Intel Analyst | Lockheed Martin

Chad Anderson

Manager, Security Intelligence Center | Reddit

splunk> .conf23

© 2023 SPLUNK INC.

Jerrod Anderson

Cyber Intel Analyst | LM CIRT Insider Threat

Chad Anderson

Manager | Reddit Security Intelligence Center

splunk> .conf23

© 2023 SPLUNK INC.

Example 1

Windows Security Event Logs

Search 1

Slow 890 Seconds

```
| search index=windows_sec user="user1" EventCode=4624
```

scanning **240,233,347** events in **889.208** seconds

Search 2

Fast 12 Seconds

```
| search index=windows_sec user="user1" "user1" EventCode=4624
```

scanning **1,043** events in **12.62** seconds

© 2023 SPLUNK INC.

Example 2

Proxy Logs

Search 1

Slow 144 Seconds

```
| search index=proxy q=* www.website.com*
```

scanning **22,437,072** events in **144.139** seconds

Search 2

Fast 37 Seconds

```
| search index=proxy TERM(q=*) TERM(www.website.com*)
```

scanning **2,896,022** events in **37.635** seconds

© 2023 SPLUNK INC.

Jerrod Anderson

Cyber Intel Analyst | LM CIRT Insider Threat

Chad Anderson

Manager | Reddit Security Intelligence Center

splunk> .conf23

© 2023 SPLUNK INC.

Example 3

Data Ingest Metrics

Search 1

Slow 21 Seconds

```
| search index=_internal source=*license_usage.log
| timechart limit=0 sum(b) by st | transpose 0
```

returned **80** results by scanning **2,621,263** events in **21.113** seconds

Search 2

Fast 1 Second

```
| tstats sum(PREFIX(b=)) as Bytes WHERE index=_internal
source=*/license_usage.log by PREFIX(st=) _time span=1d
| rename st= as SourceType  | xyseries SourceType _time Bytes
```

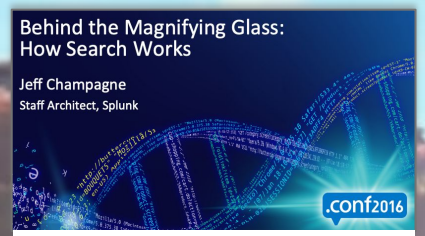
returned **27** results by scanning **2,621,263** events in **1.299** seconds

© 2023 SPLUNK INC.

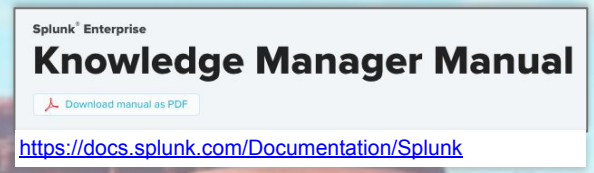
Culmination of 10 Years of .conf

Kellen Green, Clara Merriman, Richard Morgan, Martin Müller, David Veuve, Brian Wooden, Simeon Yep
And many more!

https://conf.splunk.com/files/2021/slides/TRU1133B.pdf



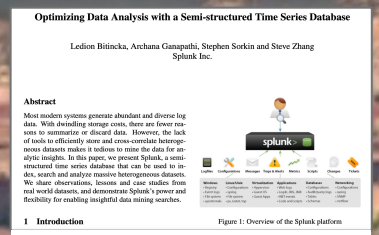
https://conf.splunk.com/files/2016/slides/behind-the-magnifying-glass-how-search-works.pdf



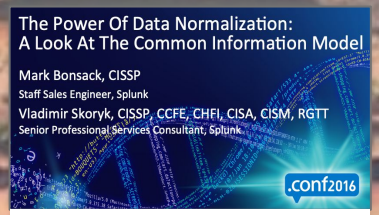
https://docs.splunk.com/Documentation/Splunk

inurl:conf.splunk.com TSTATS TERM

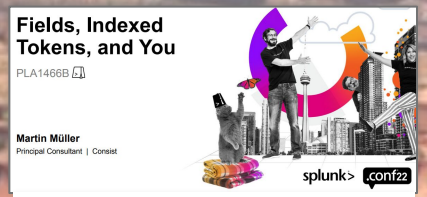
https://image-ppubs.uspto.gov/dirsearch-public/print/downloadPdf/9002854



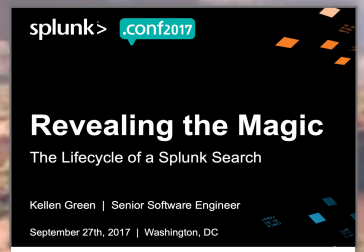
https://www.usenix.org/legacy/event/slaml10/tech/full_papers/Bitincka.pdf



https://conf.splunk.com/files/2016/slides/the-power-of-data-normalization-a-look-at-cim-under-the-hood.pdf



https://conf.splunk.com/files/2022/slides/PLA1466B.pdf



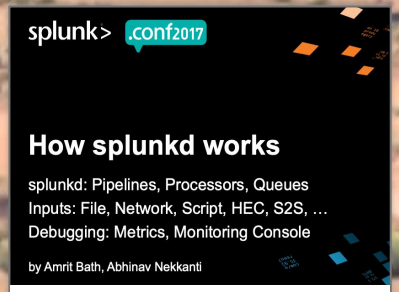
https://conf.splunk.com/files/2017/slides/revealing-the-magic-the-life-cycle-of-a-splunk-search.pdf

https://conf.splunk.com/files/2020/slides/PLA1154C.pdf

https://conf.splunk.com/files/2017/slides/observations-and-recommendations-on-splunk-performance.pdf

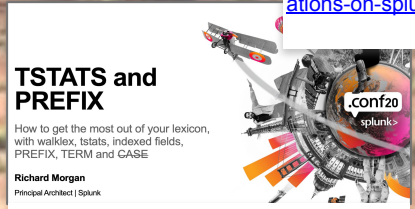
https://www.davidveuve.com/presentations.html

https://conf.splunk.com/files/2020/slides/TRU1143C.pdf



https://conf.splunk.com/files/2017/slides/how-splunkd-works.pdf

http://conf.splunk.com/files/2017/slides/searching-fast-how-to-start-using-tstats-and-other-acceleration-techniques.pdf



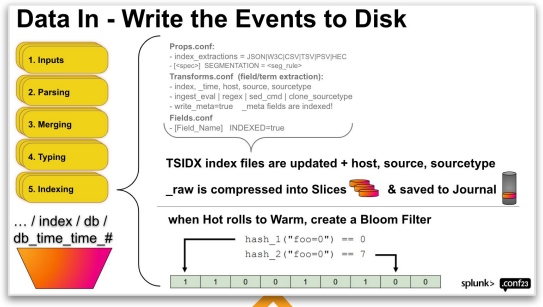
https://conf.splunk.com/files/2020/slides/PLA1089C.pdf

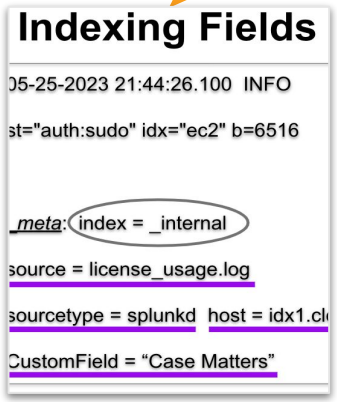
splunk> .conf23

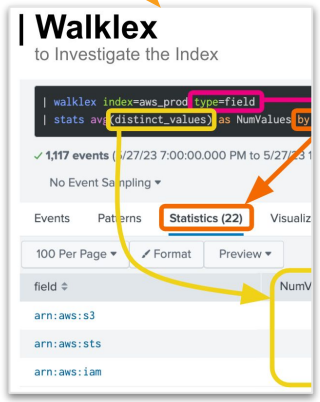
© 2023 SPLUNK INC.

Why is That Faster?

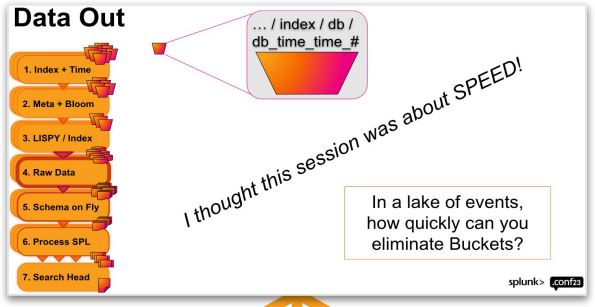
Data In

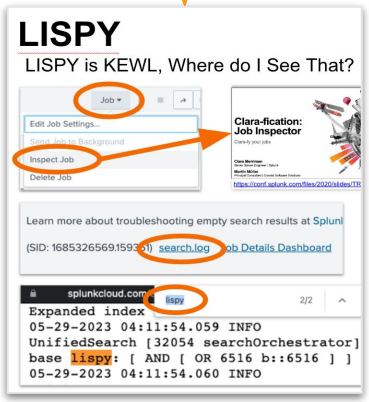


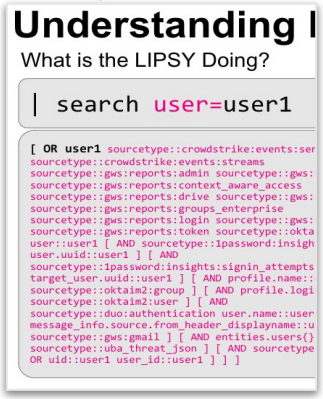




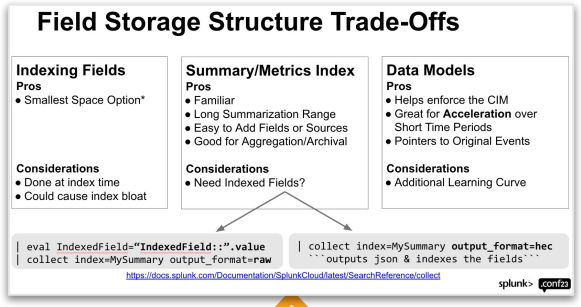
Data Out

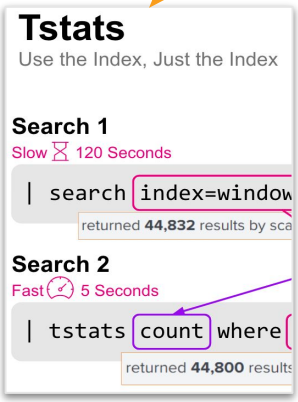


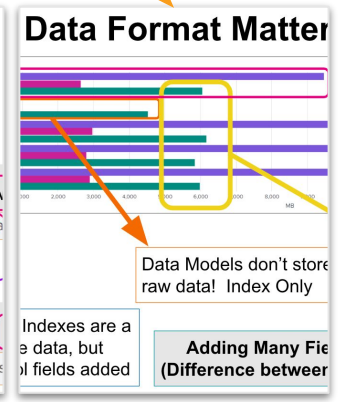




Even Faster







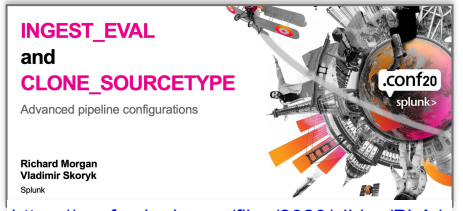
splunk> .conf23

© 2023 SPLUNK INC.

Data In

1. Inputs
2. Parsing
3. Merging
4. Typing
5. Indexing

…/ index / db / db_time_time_#



https://conf.splunk.com/files/2020/slides/PLA1154C.pdf

Props.conf:
- index_extractions = JSON|W3C|CSV|TSV|PSV|HEC
- [<spec>] SEGMENTATION = <seg_rule>

Transforms.conf (field/term extraction):
- index, _time, host, source, sourcetype
- ingest_eval | regex | sed_cmd | clone_sourcetype
- write_meta=true _meta fields are indexed!

Fields.conf
- [Field_Name] INDEXED=true

Time Series Index Files (TSIDX) are being created

```
05-25-2023 21:44:26.100 INFO

st="auth:sudo" idx="ec2" b=6516
```

© 2023 SPLUNK INC.

Indexing Fields

05-25-2023 21:44:26.100 INFO

st="auth:sudo" idx="ec2" b=6516

_meta: index = _internal

source = license_usage.log

sourcetype = splunkd host = idx1.cloud.com

CustomField = “Case Matters”

EarliestEventTime_LatestEventTime.tsidx

CustomField::Case Matters
customfield::case matters
host::idx1.cloud.com
source::license_usage.log
sourcetype::splunkd

© 2023 SPLUNK INC.

Major Breakers

05-25-2023 21:44:26.100 INFO

st="auth:sudo" idx="ec2" b=6516

_meta: index = _internal

source = license_usage.log

sourcetype = splunkd host = idx1.cloud.com

CustomField = “Case Matters”

EarliestEventTime_LatestEventTime.tsidx

CustomField::Case Matters
customfield::case matters
host::idx1.cloud.com
source::license_usage.log
sourcetype::splunkd
05-25-2023
21:44:26.100
auth:sudo
b=6516
ec2
idx=
info
st=

© 2023 SPLUNK INC.

Minor Breakers

05-25-2023 21:44:26.100 INFO

st="auth:sudo" idx="ec2" b=6516

_meta: index = _internal

source = license_usage.log

sourcetype = splunkd host = idx1.cloud.com

CustomField = “Case Matters”

EarliestEventTime_LatestEventTime.tsidx

CustomField::Case Matters
customfield::case matters
host::idx1.cloud.com
source::license_usage.log
sourcetype::splunkd
05
05-25-2023
100
2023
21
21:44:26.100
25
26
44
6516
auth
auth:sudo
b
b=6516
ec2
idx
idx=
info
st
st=
sudo

© 2023 SPLUNK INC.

Back to Example 3

05-25-2023 21:44:26.100 INFO

st="auth:sudo" idx="ec2" b=6516

```
| search index=_internal source=license_usage.log
| timechart limit=0 sum(b) by st
```

```
| tstats sum(PREFIX(b=)) as Bytes by PREFIX(st=)
```

EarliestEventTime_LatestEventTime.tsidx

CustomField::Case Matters
customfield::case matters
host::idx1.cloud.com
source::license_usage.log
sourcetype::splunkd
05
05-25-2023
100
2023
6516
auth
auth:sudo
b
b=6516
ec2
st
st=
sudo

© 2023 SPLUNK INC.

Back to Example 3

05-25-2023 21:44:26.100 INFO

st="auth:sudo" idx="ec2" b=6516

```
| sear
| time
```

Major Breakers impact how you use the Index!

```
| tstats sum(PREFIX(b=)) as Bytes by PREFIX(st=)
```

EarliestEventTime_LatestEventTime.tsidx

CustomField::Case Matters
customfield::case matters
host::idx1.cloud.com
source::license_usage.log
sourcetype::splunkd
05
05-25-2023
100
2023
auth:sudo
b
b=6516
ec2
st=
sudo

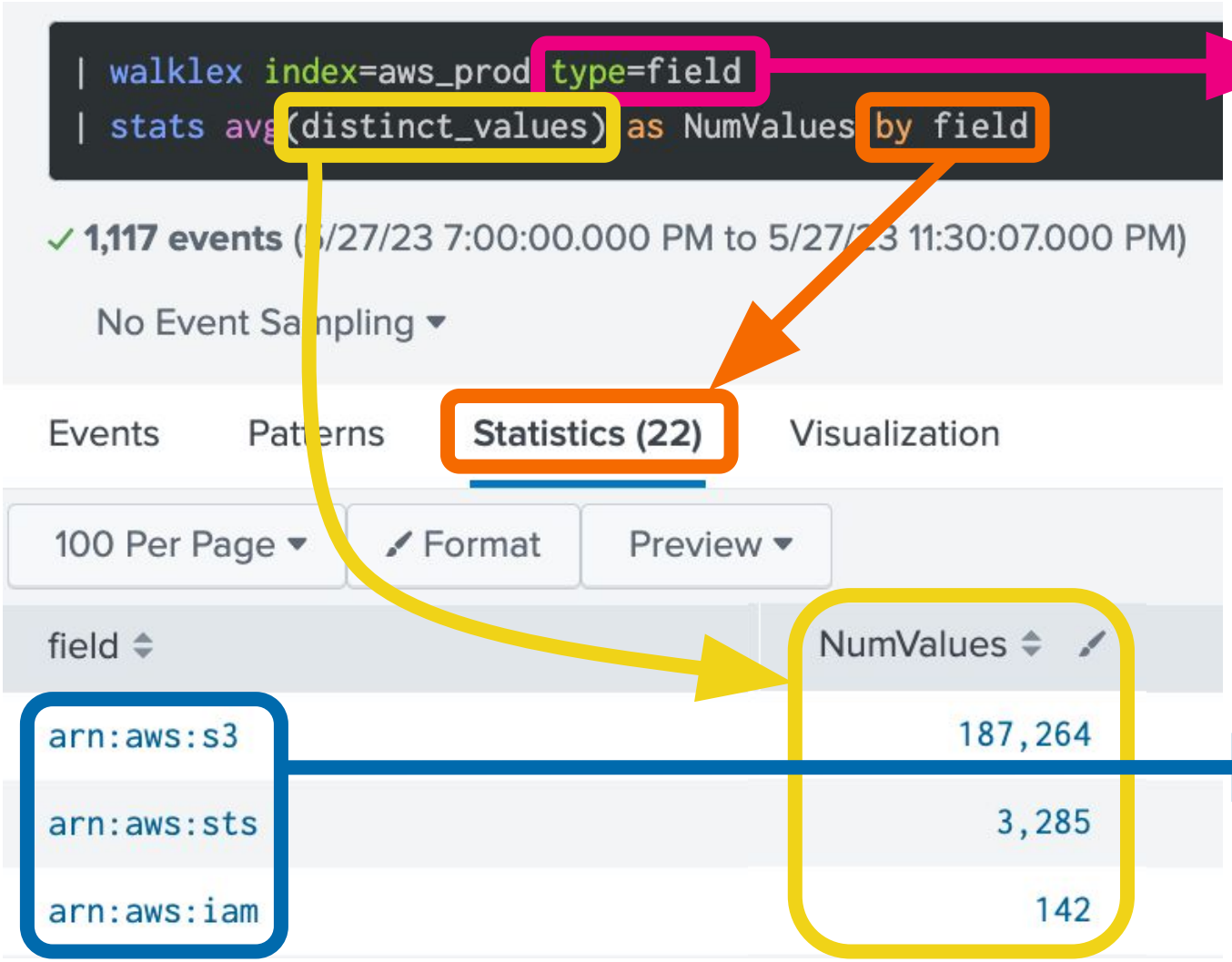
© 2023 SPLUNK INC.

| Walklex

to Investigate the Index

Splunk® Enterprise
Knowledge Manager Manual
Download manual as PDF

https://docs.splunk.com/Documentation/SplunkCloud/latest/SearchReference/walklex



Type:

- field
- fieldvalue
- term
- all

Are these fields intentional?

splunk> .conf23

© 2023 SPLUNK INC.

| Walklex

to Investigate the Index Bloat

Pivot from field to:
type=fieldvalue

Splunk® Enterprise
Knowledge Manager Manual
Download manual as PDF

https://docs.splunk.com/Documentation/SplunkCloud/latest/SearchReference/walklex

```
| walklex index=aws_prod type=fieldvalue pattern=arn:aws:s3*
| stats avg(count) as NumEvents by term
```

✓ **100,528 events** (5/27/23 11:34:00.000 PM to 5/27/23 11:54:47.000 PM)

No Eve

Events

100 Per Page ▾ | Format | Preview ▾

term	NumEvents
arn:aws:s3:::reddit-uploaded-media	17,984
arn:aws:s3:::reddit-hosted-video	13,823
arn:aws:s3:::reddit-hosted-media	11,650
arn:aws:s3:::reddit-subreddit-uploaded-media	2,884

Pattern:

- Filter the search
- Accepts Wildcards

Multi-Colon Data can Bloat the Index!

“url”=“www.web.com?Ad:1234::5678::abcd”

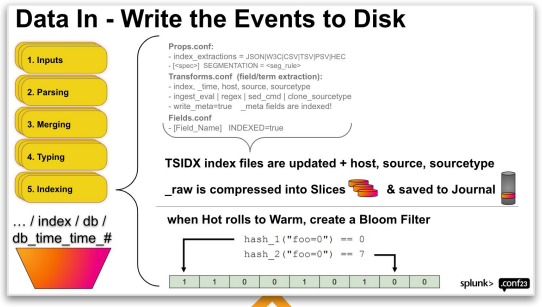
Multi-Colon Data?

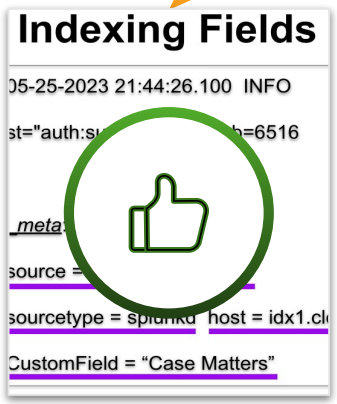
- CloudTrail
- Proxy Logs
- etc

© 2023 SPLUNK INC.

Why is That Faster?

Data In





splunk> .conf23

© 2023 SPLUNK INC.

Data In - Write the Events to Disk

1. Inputs
2. Parsing
3. Merging
4. Typing
5. Indexing

Props.conf:
- index_extractions = JSON|W3C|CSV|TSV|PSV|HEC
- [<spec>] SEGMENTATION = <seg_rule>

Transforms.conf (field/term extraction):
- index, _time, host, source, sourcetype
- ingest_eval | regex | sed_cmd | clone_sourcetype
- write_meta=true _meta fields are indexed!

Fields.conf
- [Field_Name] INDEXED=true

TSIDX index files are updated + host, source, sourcetype

_raw is compressed into Slices  **within the Journal** 

© 2023 SPLUNK INC.

Data In - Create the Bloom Filter

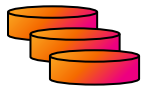
1. Inputs
2. Parsing
3. Merging
4. Typing
5. Indexing

Props.conf:
- index_extractions = JSON|W3C|CSV|TSV|PSV|HEC
- [<spec>] SEGMENTATION = <seg_rule>

Transforms.conf (field/term extraction):
- index, _time, host, source, sourcetype
- ingest_eval | regex | sed_cmd | clone_sourcetype
- write_meta=true _meta fields are indexed!

Fields.conf
- [Field_Name] INDEXED=true

TSIDX index files are updated + host, source, sourcetype

_raw is compressed into Slices  **within the Journal** 

---

when Bucket is made Warm, create the Bloom Filter

```
hash_1("foo=0") == 0
hash_2("foo=0") == 7
```

1	1	0	0	1	0	1	0	0

AUTOMATICALLY GENERATED

splunk> .conf23

© 2023 SPLUNK INC.

Data In X 10000s of Buckets

1. Inputs
2. Parsing
3. Merging
4. Typing
5. Indexing

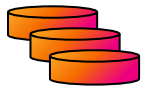
… / index / db / db_time_time_#

Props.conf:
- index_extractions = JSON|W3C|CSV|TSV|PSV|HEC
- [<spec>] SEGMENTATION = <seg_rule>

Transforms.conf (field/term extraction):
- index, _time, host, source, sourcetype
- ingest_eval | regex | sed_cmd | clone_sourcetype
- write_meta=true _meta fields are indexed!

Fields.conf
- [Field_Name] INDEXED=true

TSIDX index files are updated + host, source, sourcetype

_raw is compressed into Slices  **within the Journal** 

when Bucket is made Warm, create the Bloom Filter

```
hash_1("foo=0") == 0
hash_2("foo=0") == 7
```

1	1	0	0	1	0	1	0	0

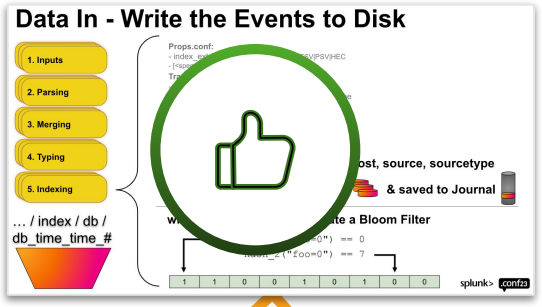
AUTOMATICALLY GENERATED

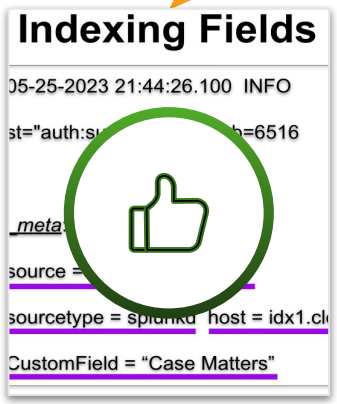
splunk> .conf23

© 2023 SPLUNK INC.

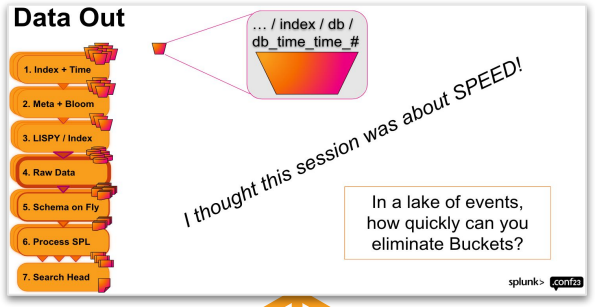
Why is That Faster?

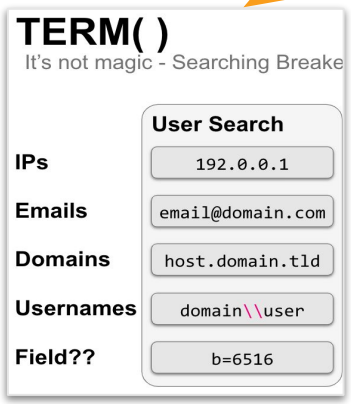
Data In

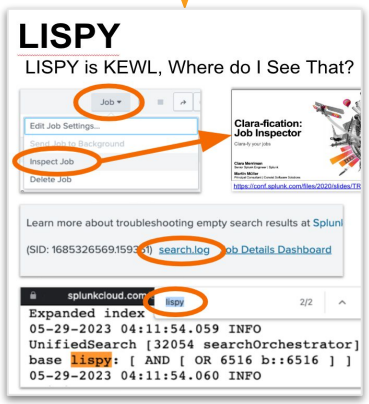


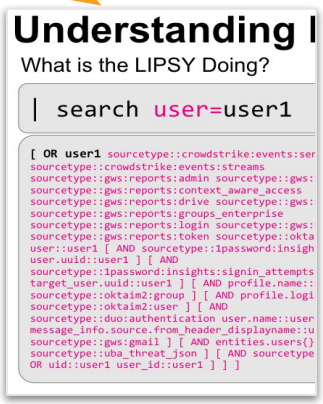


Data Out





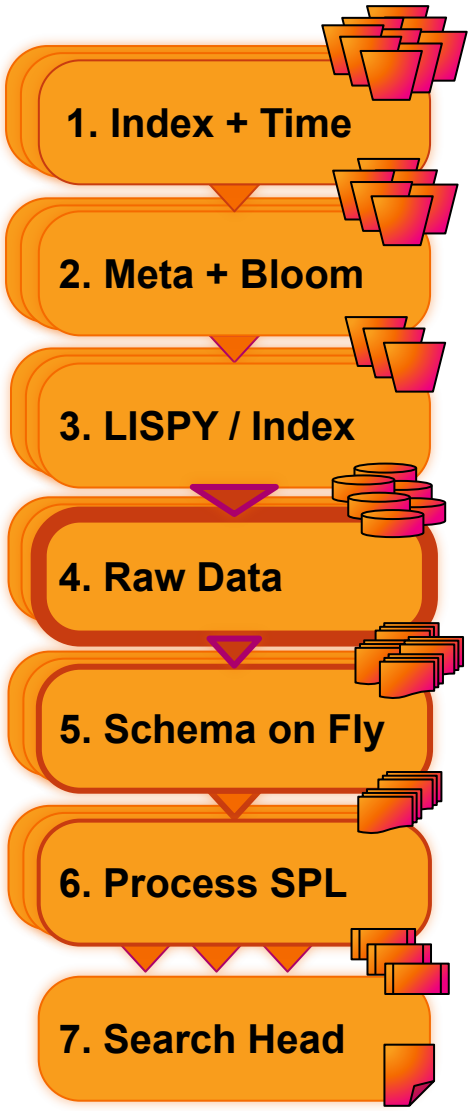


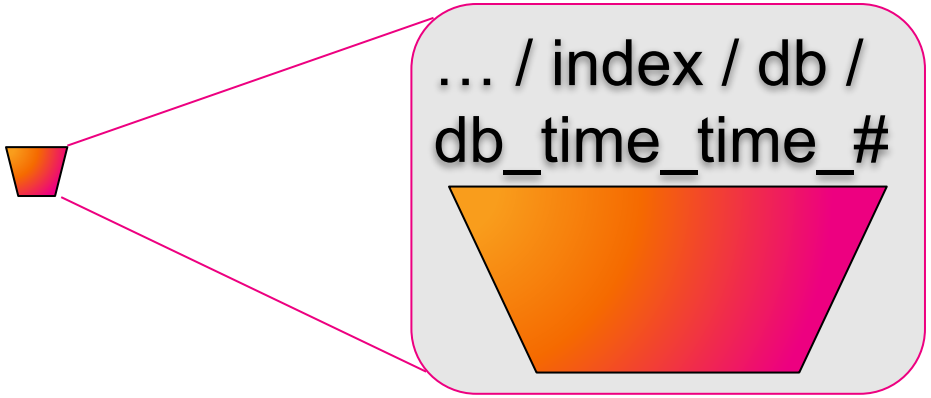


splunk> .conf23

© 2023 SPLUNK INC.

Data Out





I thought this session was about SPEED!

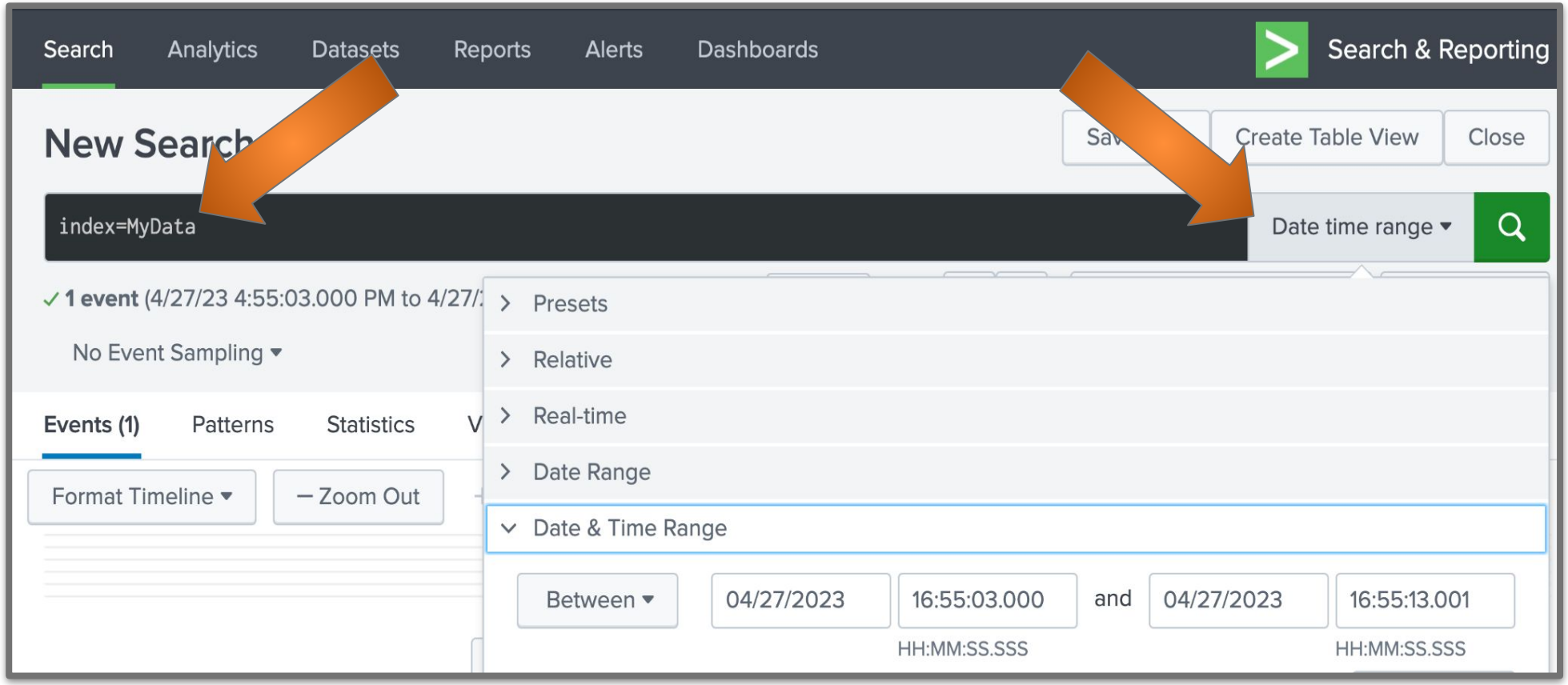
In a lake of events, how quickly can you eliminate Buckets?

© 2023 SPLUNK INC.

Data Out - 1. Index + Time

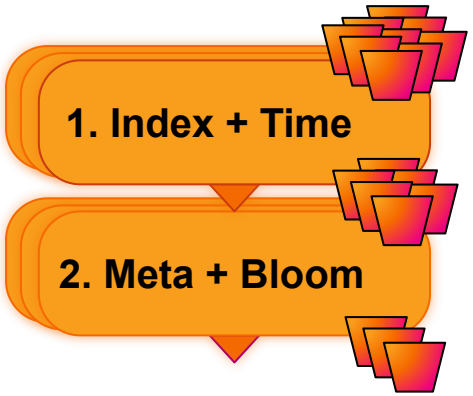
FREE

First Bucket Reduction



© 2023 SPLUNK INC.

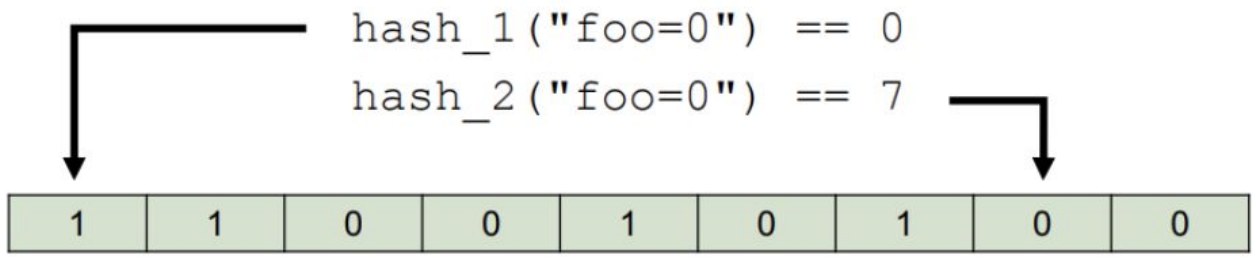
Data Out - 2. Meta (H, S, ST) + Bloom Filters



FREE

Eliminate More Buckets!

Bloom Filters answer: does term "`foo=0`" exit?
Can guarantee that term does NOT exist in any events = SKIP

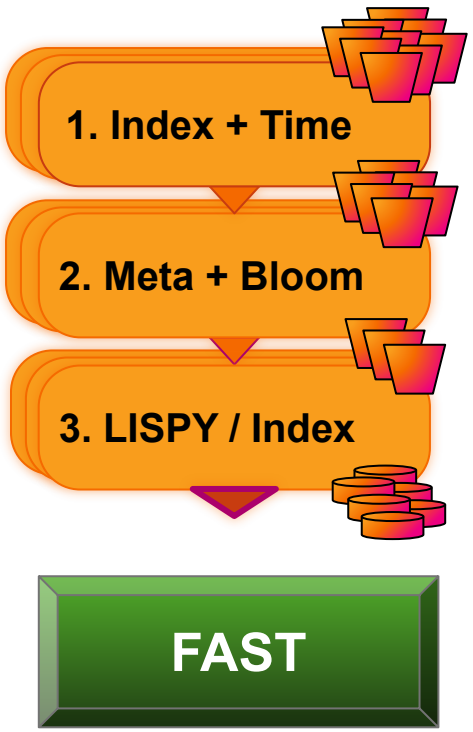


But cannot guarantee term does exist ~98% accurate

Bloom Filters **DO NOT** use Wildcards*!

© 2023 SPLUNK INC.

Data Out - 3. Search the Index



EarliestEventTime_LatestEventTime.tsidx

CustomField::Case Matters
customfield::case matters
host::idx1.cloud.com
source::license_usage.log
sourcetype::splunkd
05
05-25-2023
100
2023
21
21:44:26.100
25
26
44
6516
auth
auth:sudo
b
b=6516
ec2
idx
idx=
info
st
st=
sudo

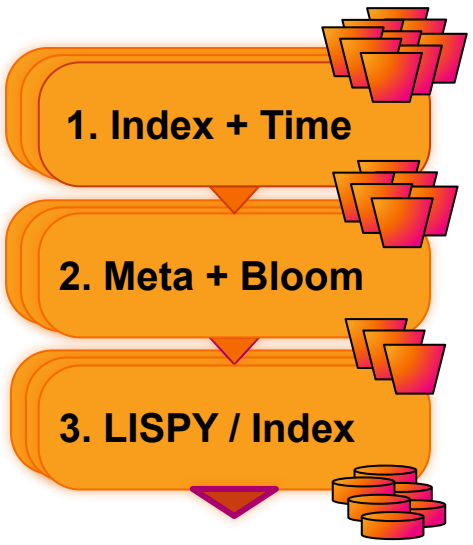
TSIDX files are FAST! But not Free.

TSIDX points to specific event(s)

An event is stored in a Slice 

© 2023 SPLUNK INC.

Data Out - 3. Search the Index



EarliestEventTime_LatestEventTime.tsidx

CustomField::Case Matters
customfield::case matters
host::idx1.cloud.com
source::license_usage.log
sourcetype::splunkd
05
05-25-2023
100
2023
21
21:44:26.100
25
26
44

b=6516

st=
sudo

TSIDX files are FAST! But not Free.

TSIDX points to specific event(s)

An event is stored in a Slice 

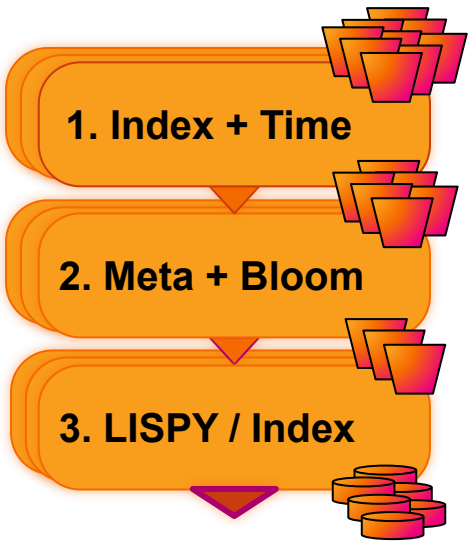
```
| search index=_internal b=6516
```

```
[ AND index::_internal [ OR 6516 b::6516 ] ]
```

splunk> .conf23

© 2023 SPLUNK INC.

Data Out - 3. Search the Index with TERM



EarliestEventTime_LatestEventTime.tsidx

CustomField::Case Matters
customfield::case matters
host::idx1.cloud.com
source::license_usage.log
sourcetype::splunkd
05
05-25-2023
100
2023
21
21:44:26.100
25
26
44

```
| search index=_internal TERM(b=6516)
```

b=6516
ec2

```
[ AND index::_internal b=6156 ]
```

st
st=
sudo

TSIDX files are FAST! But not Free.

TSIDX points to specific event(s)

An event is stored in a Slice 

splunk> .conf23

© 2023 SPLUNK INC.

TERM( )

It's not magic!

Wildcards

Minor Breakers

Search 1

Slow 144 Seconds

```
| search index=proxy q=* www.website.com*
```

scanning **22,437,072** events in **144.139** seconds

Search 2

Fast 37 Seconds

```
| search index=proxy TERM(q=*) TERM(www.website.com*)
```

scanning **2,896,022** events in **37.635** seconds

© 2023 SPLUNK INC.

TERM( )

It's not magic - Searching Breakers

Minor Breakers

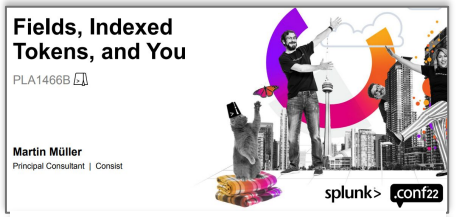
/ \ : = @ . - _ $ # %

	User Search	Base Lispy	TERM( ) Lispy
IPs	192.0.0.1	[AND 0 1 192]	192.0.0.1
Emails	email@domain.com	[AND com domain email]	email@domain.com
Domains	host.domain.tld	[AND domain host tld]	host.domain.tld
Usernames	domain\\user	[AND domain user]	domain\user
Field??	b=6516	[AND 6156 b::6516]	b=6516

© 2023 SPLUNK INC.

TERM( )

It's not magic - Searching Wildcards



https://conf.splunk.com/files/2022/slides/PLA1466B.pdf

	User Search	Base Lispy	TERM( ) Lispy
Trailing*	SPLUN*	splun*	splun*
*Leading	*PLUNK	*ALL THE EVENTS*	*plunk
Cont-ains	*OMAIN\\USE*	[AND use*]	*omain\use*
Field = *	B=*	*ALL THE EVENTS*	b=*
Field* =	B*=6516	[OR 6156 b*::6156]	b*=6516

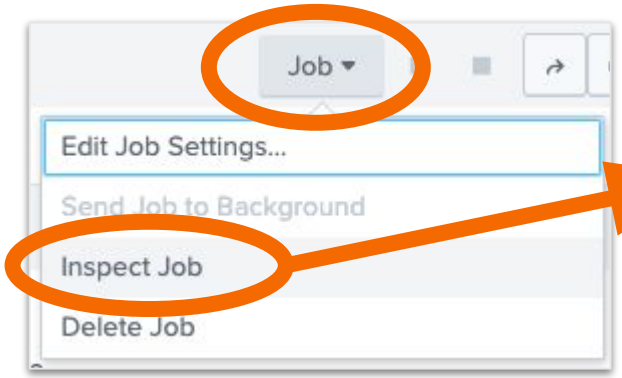
Do Bloom Filters Support Wildcards? NO

splunk> .conf23

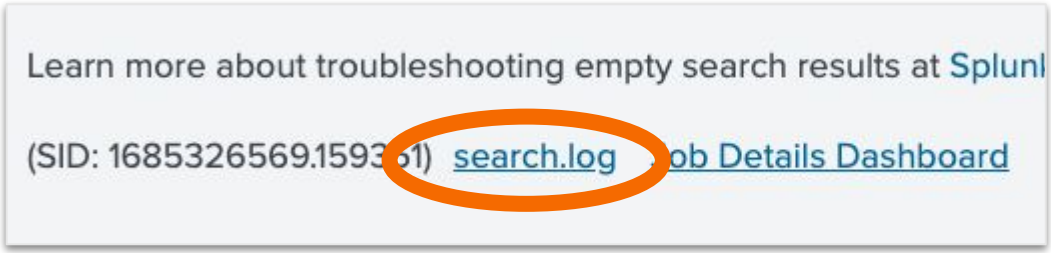
© 2023 SPLUNK INC.

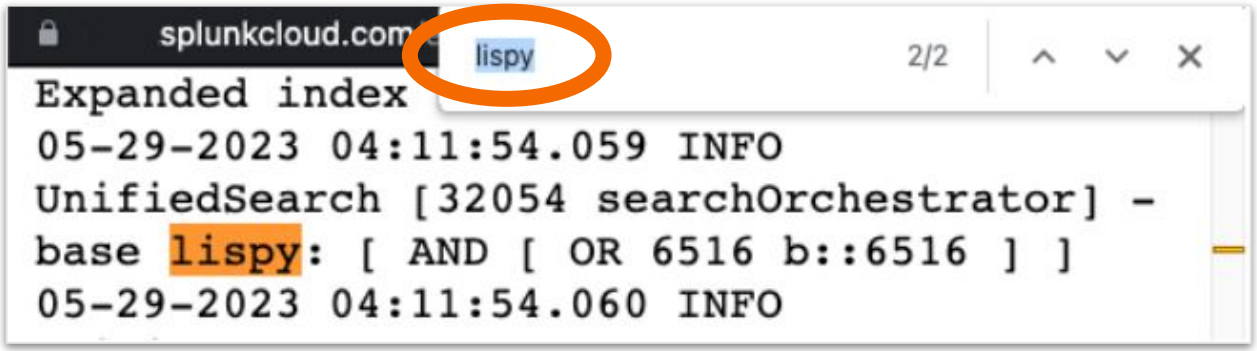
LISPY

LISPY is KEWL, Where Do I See That?



https://conf.splunk.com/files/2020/slides/TRU1143C.pdf

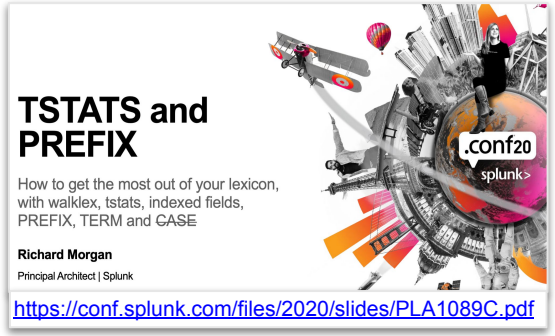




Search Performance Evaluator Dashboard

Search specification

Name	Description	Value
SPL	The search entered by the user	search b=6516
Optimized SPL	The search post optimization	| search b=6516
Keywords	The keywords found in SPL	b::6516
LISPY	The query on TSIDX	[AND [OR 6516 b::6516]]



https://conf.splunk.com/files/2020/slides/PLA1089C.pdf

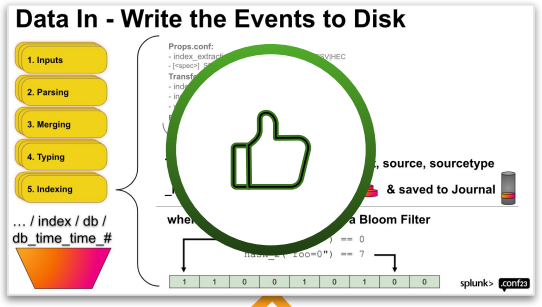
https://github.com/silkyrich/cluster_health_tools/blob/master/default/data/ui/views/search_performance_evaluator.xml

splunk> .conf23

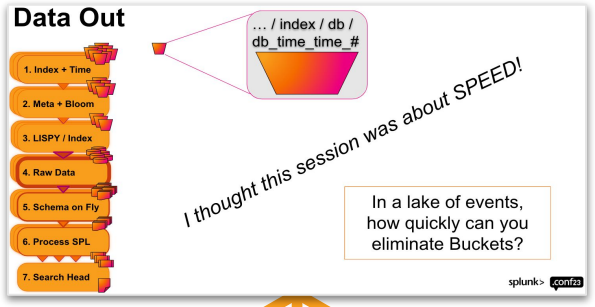
© 2023 SPLUNK INC.

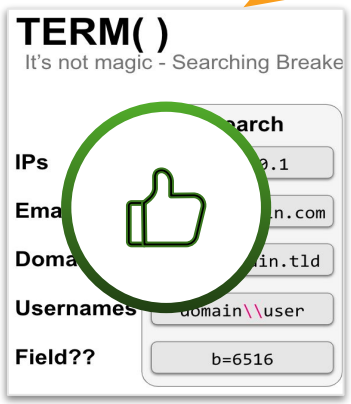
Why is That Faster?

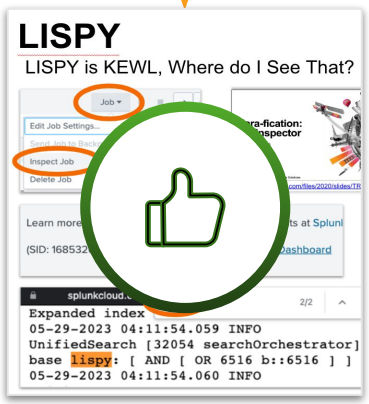
Data In

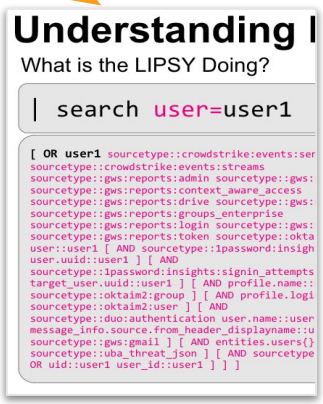


Data Out









splunk> .conf23

© 2023 SPLUNK INC.

Example 1

Windows Security Event Logs

Search 1

Slow 890 Seconds

```
| search index=windows_sec user="user1" EventCode=4624
```

scanning **240,233,347** events in **889.208** seconds

Search 2

Fast 12 Seconds

```
| search index=windows_sec user="user1" "user1" EventCode=4624
```

scanning **1,043** events in **12.62** seconds

© 2023 SPLUNK INC.

Understanding Example 1

What is the LISPY Doing?

```
| search user=user1
```

LISPY

```
[ OR user1 sourcetype::crowdstrike:events:sensor
sourcetype::crowdstrike:events:streams
sourcetype::gws:reports:admin sourcetype::gws:reports:calendar
sourcetype::gws:reports:context_aware_access
sourcetype::gws:reports:drive sourcetype::gws:reports:gcp
sourcetype::gws:reports:groups_enterprise
sourcetype::gws:reports:login sourcetype::gws:reports:saml
sourcetype::gws:reports:token sourcetype::oktaim2:log user1
user::user1 [ AND sourcetype::1password:insights:item_usages
user.uuid::user1 ] [ AND
sourcetype::1password:insights:signin_attempts
target_user.uuid::user1 ] [ AND profile.name::user1
sourcetype::oktaim2:group ] [ AND profile.login::user1
sourcetype::oktaim2:user ] [ AND sourcetype::duo:authentication
user.name::user1 ] [ AND
message_info.source.from_header_displayname::user1
sourcetype::gws:gmail ] [ AND entities.users{}.name::user1
sourcetype::uba_threat_json ] [ AND sourcetype::audittrail [ OR
uid::user1 user_id::user1 ] ] ]
```

3 Possible Solutions:

```
| search user=user1 user1
```

LISPY

```
[ AND user1 ]
```

```
| search TERM(user=user1)
```

LISPY

```
[ AND user=user1 ]
```

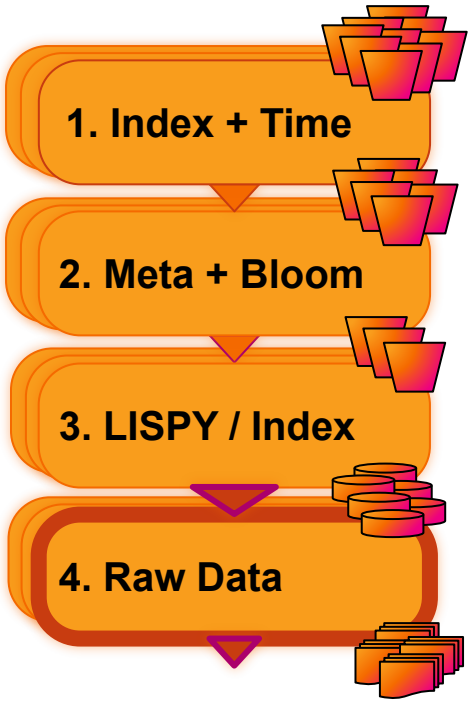
```
| search user::user1
```

LISPY

```
[ AND user::user1 ]
```

© 2023 SPLUNK INC.

Data Out - 4. Read Compressed Data



SLOW

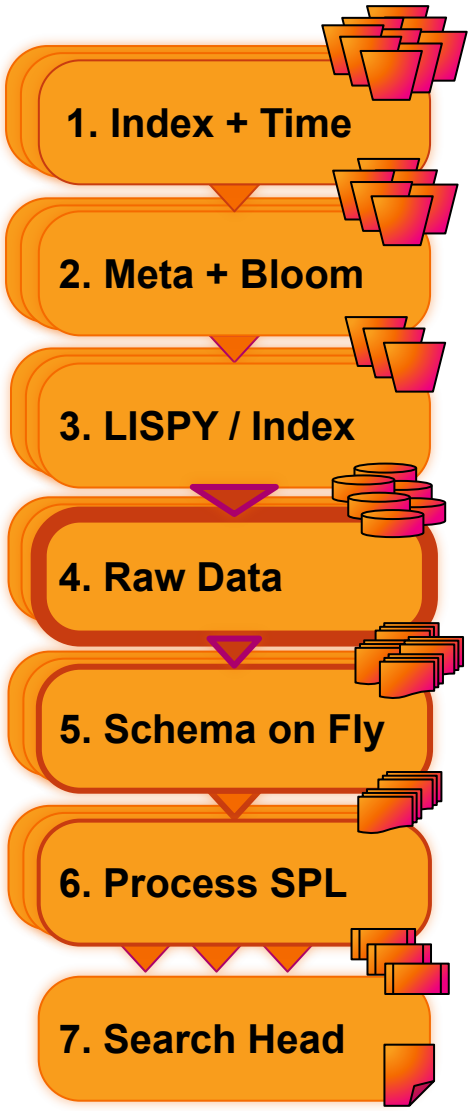
Extract Slice of Compressed Raw Data

Decompress Events (CPU Expensive)

Extract Specific Events (too many is Memory Expensive)

© 2023 SPLUNK INC.

Data Out - 5.Parse, 6.SPL, 7.Search Head



5. Schema on the Fly - parse fields

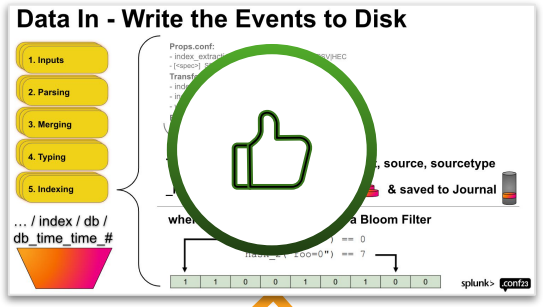
6. Streaming SPL™ Commands

7. Return required data to the Search Head
   Verbose or Fast?

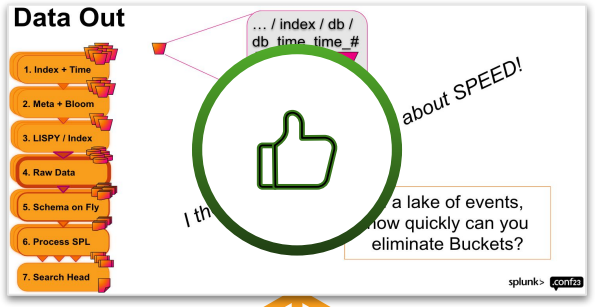
© 2023 SPLUNK INC.

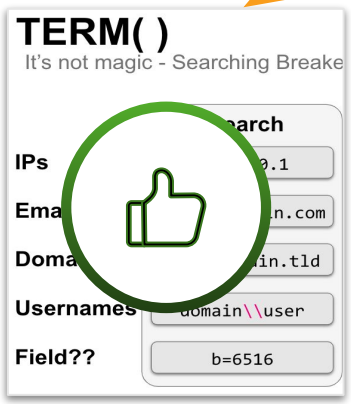
Why is That Faster?

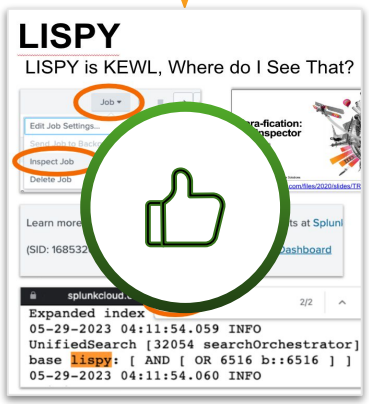
Data In

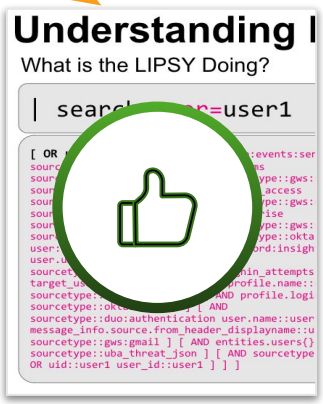


Data Out

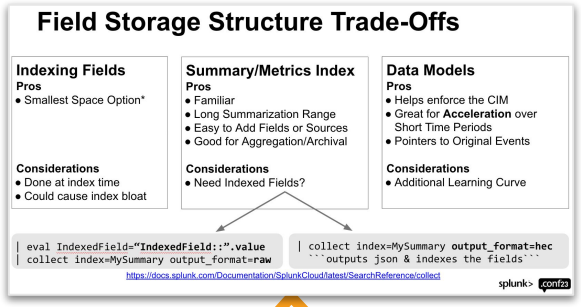


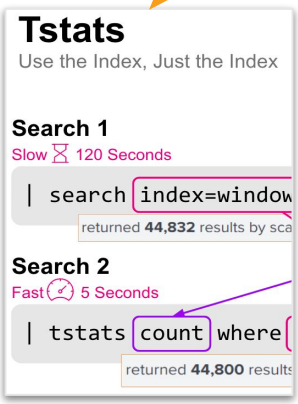


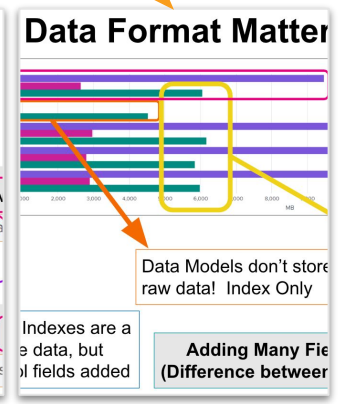




Even Faster





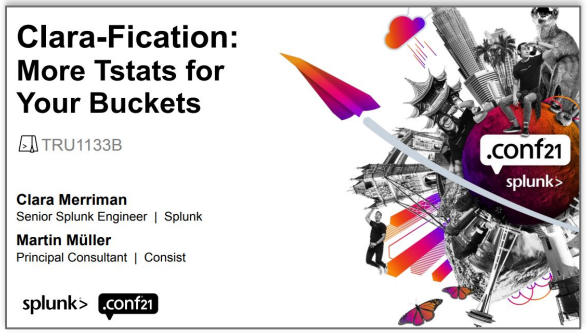


splunk> .conf23

© 2023 SPLUNK INC.

Tstats

Use the Index, Just the Index



https://conf.splunk.com/files/2021/slides/TRU1133B.pdf

Search 1

Slow 120 Seconds

```
| search index=windows_security | stats count by source
```

returned **44,832** results by scanning **196,540,296** events in **120.761** seconds

Search 2

Fast 5 Seconds

```
| tstats count where index=windows_security by source
```

returned **44,800** results by scanning **196,407,587** events in **4.557** seconds

© 2023 SPLUNK INC.

Tstats Prefix( )

Use the Index, Just the Index for any TERM

```
| tstats sum(PREFIX(b=)) as Bytes WHERE index=_internal
source=*/license_usage.log by PREFIX(05-) _time span=1d
```

CustomField::Case Matters
customfield::case matters
host::idx1.cloud.com
05-25-2023
auth:sudo
b=6516
ec2
idx=
st=

- PREFIX( ) can use any TERM in the index
- Want to count bytes by terms that start with 05?

05-
30-2023

- Major Breakers are not indexed: {“json”:“value”}
  - That is why PREFIX(st=) didn’t work

- | tstats has SPAN built in!

© 2023 SPLUNK INC.

Field Storage Structure Trade-Offs

Add Fields to Index

Pros

- Smallest Space Option*

Considerations

- Done at index time
- Could cause index bloat

Summary/Metrics Index

Pros

- Familiar
- Long Summarization Range
- Easy to Add Fields or Sources
- Good for Aggregation/Archival

Considerations

- Only Copy Required Data
- Need Indexed Fields?

Data Models

Pros

- Helps enforce the CIM
- Great for **Acceleration** over Short Time Periods
- Pointers to Original Events

Considerations

- Additional Learning Curve

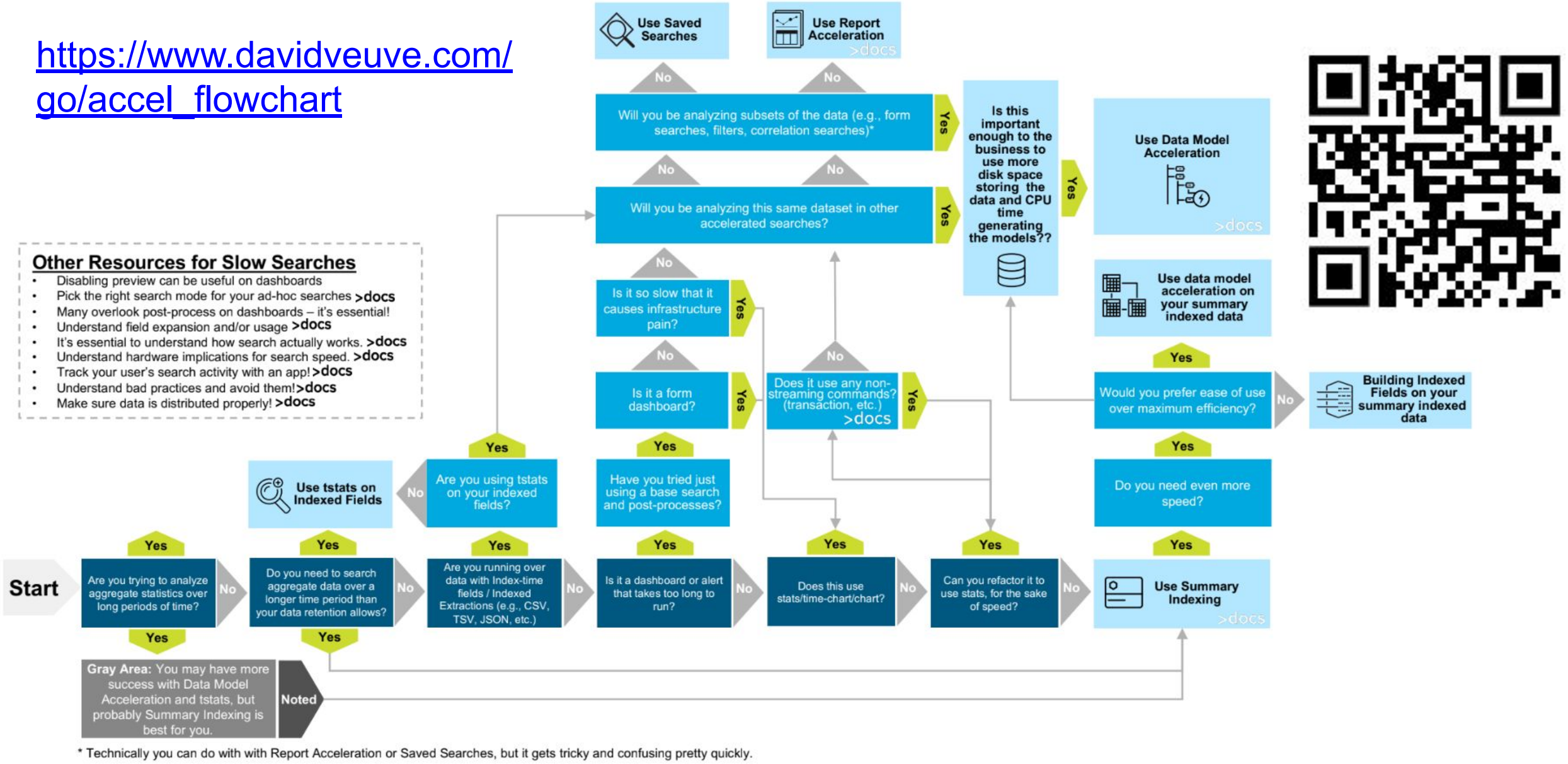
```
| eval IndexedField="IndexedField::".value
| collect index=MySummary output_format=raw
```

```
| collect index=MySummary output_format=hec
  ```outputs json & indexes the fields```
```

https://docs.splunk.com/Documentation/SplunkCloud/latest/SearchReference/collect
```

© 2023 SPLUNK INC.

https://www.davidveuve.com/go/accel_flowchart



Other Resources for Slow Searches

- Disabling preview can be useful on dashboards
- Pick the right search mode for your ad-hoc searches >docs
- Many overlook post-process on dashboards – it's essential!
- Understand field expansion and/or usage >docs
- It's essential to understand how search actually works. >docs
- Understand hardware implications for search speed. >docs
- Track your user's search activity with an app! >docs
- Understand bad practices and avoid them! >docs
- Make sure data is distributed properly! >docs

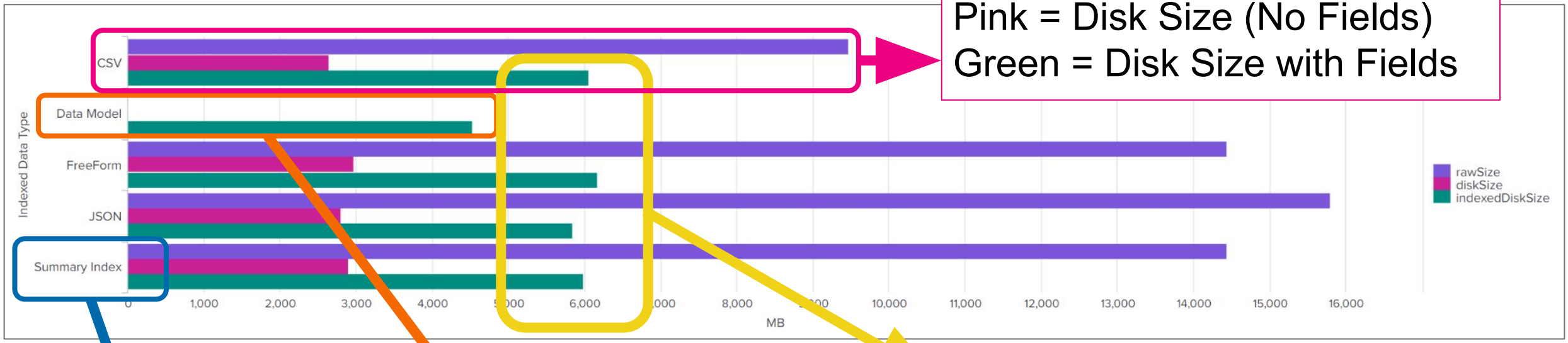
* Technically you can do with with Report Acceleration or Saved Searches, but it gets tricky and confusing pretty quickly.

splunk> .conf23

© 2023 SPLUNK INC.

Does Data Format Matter?

Purple = Raw Event Size
Pink = Disk Size (No Fields)
Green = Disk Size with Fields



Data Models don't store raw data! Index Only

Total disk is about the same no matter the data format

Summary Indexes are a copy of the data, but you control fields added

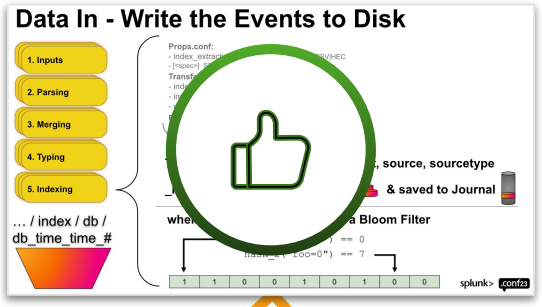
Adding Many Fields Causes Bloat (Difference between Pink & Green Bars)

splunk> .conf23

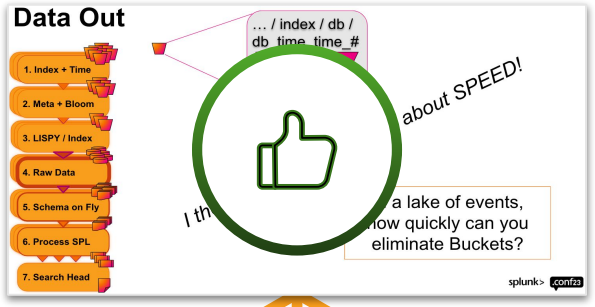
© 2023 SPLUNK INC.

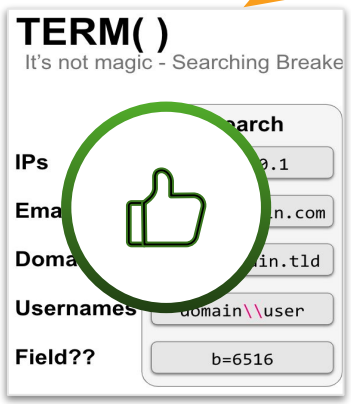
Why is That Faster?

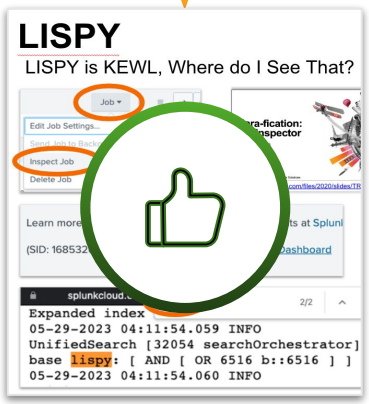
Data In

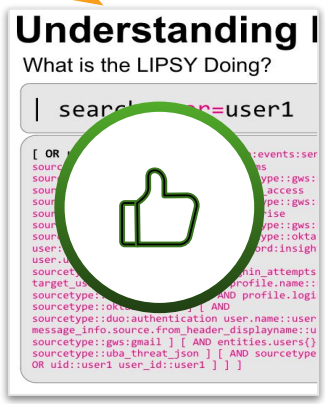


Data Out

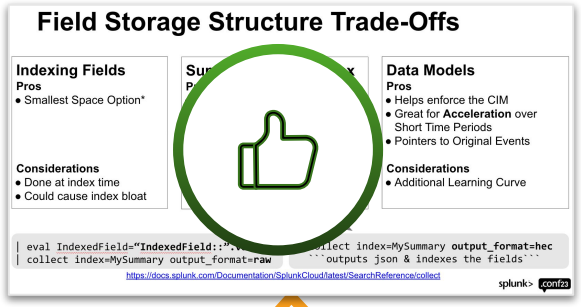


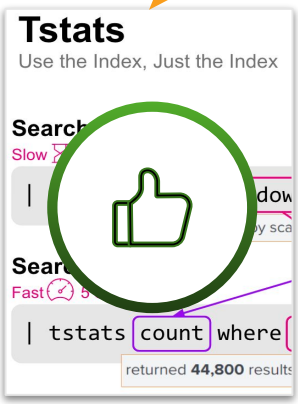


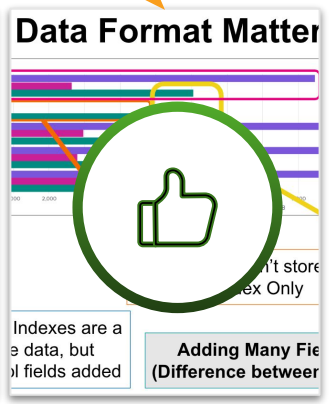




Even Faster







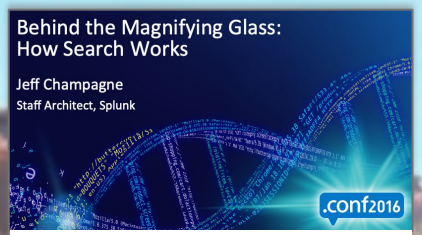
splunk> .conf23

© 2023 SPLUNK INC.

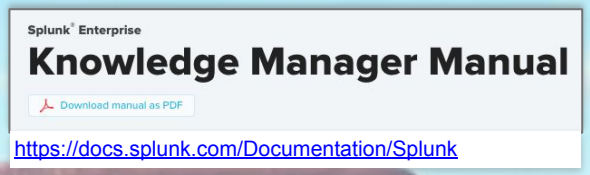
Culmination of 10 years of .conf

Kellen Green, Clara Merriman, Richard Morgan, Martin Müller, David Veuve, Brian Wooden, Simeon Yep
And many more!

https://conf.splunk.com/files/2021/slides/TRU1133B.pdf



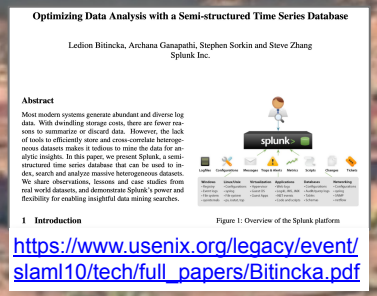
https://conf.splunk.com/files/2016/slides/behind-the-magnifying-glass-how-search-works.pdf



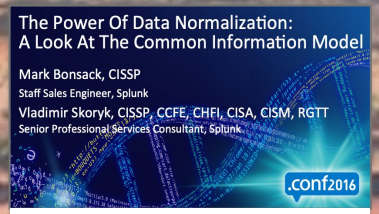
https://docs.splunk.com/Documentation/Splunk

inurl:conf.splunk.com TSTATS TERM

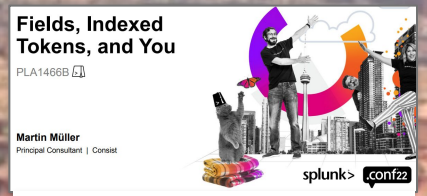
https://image-ppubs.uspto.gov/dirsearch-public/print/downloadPdf/9002854



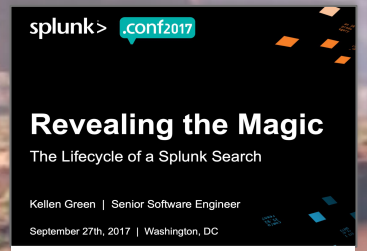
https://www.usenix.org/legacy/event/slaml10/tech/full_papers/Bitincka.pdf



https://conf.splunk.com/files/2016/slides/the-power-of-data-normalization-a-look-at-cim-under-the-hood.pdf

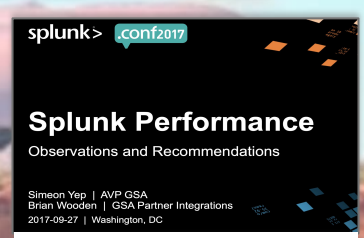


https://conf.splunk.com/files/2022/slides/PLA1466B.pdf



https://conf.splunk.com/files/2017/slides/revealing-the-magic-the-life-cycle-of-a-splunk-search.pdf

https://conf.splunk.com/files/2020/slides/PLA1154C.pdf



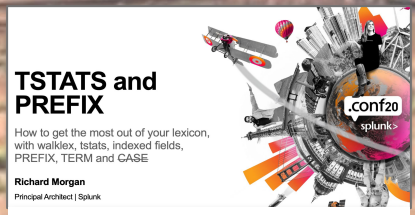
https://conf.splunk.com/files/2017/slides/observations-and-recommendations-on-splunk-performance.pdf

https://www.davidveuve.com/presentations.html

https://conf.splunk.com/files/2020/slides/TRU1143C.pdf

https://conf.splunk.com/files/2017/slides/how-splunkd-works.pdf

http://conf.splunk.com/files/2017/slides/searching-fast-how-to-start-using-tstats-and-other-acceleration-techniques.pdf



https://conf.splunk.com/files/2020/slides/PLA1089C.pdf

splunk> .conf23

© 2023 SPLUNK INC.

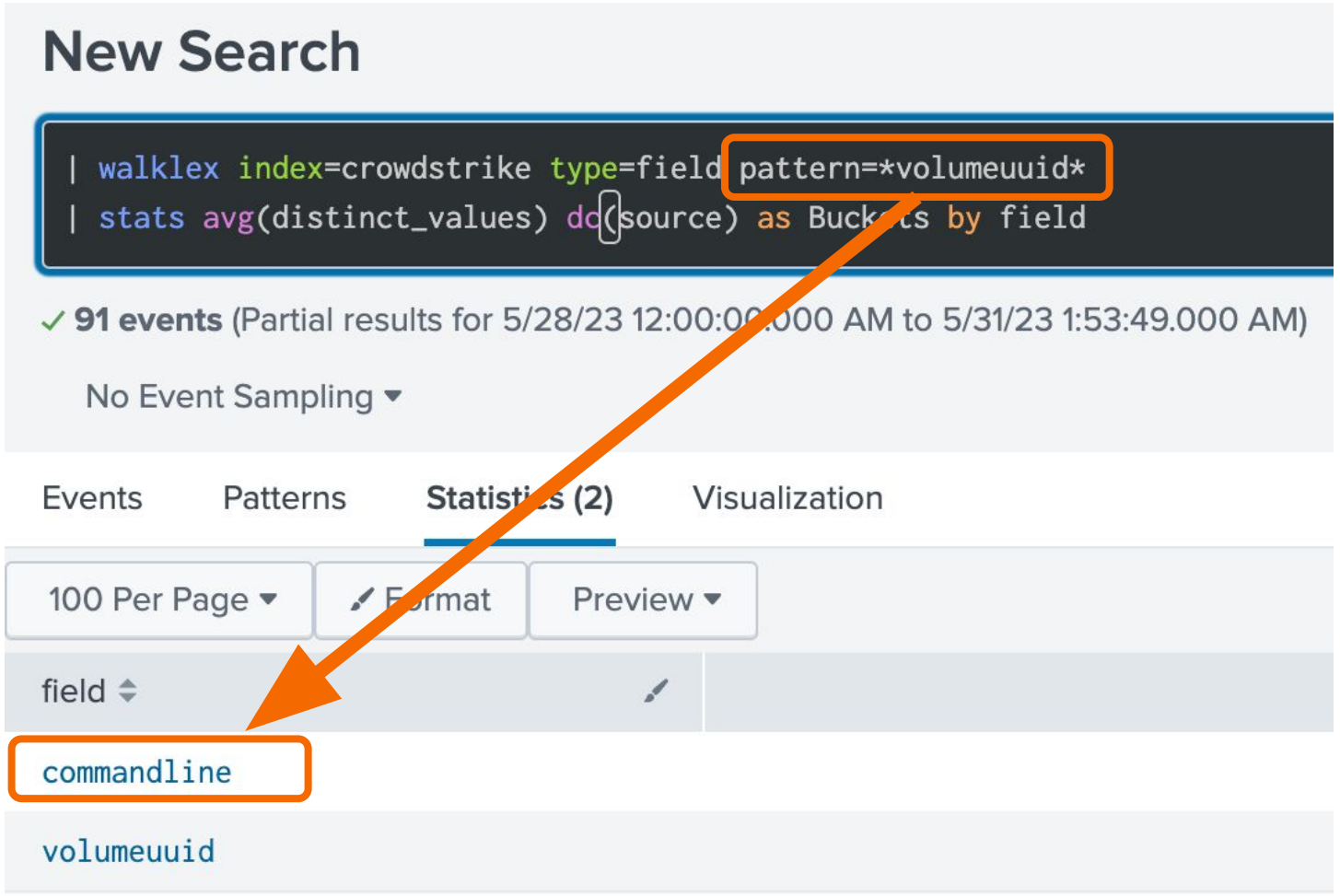
Thank You

splunk> .conf23

© 2023 SPLUNK INC.

Bugs in Walklex

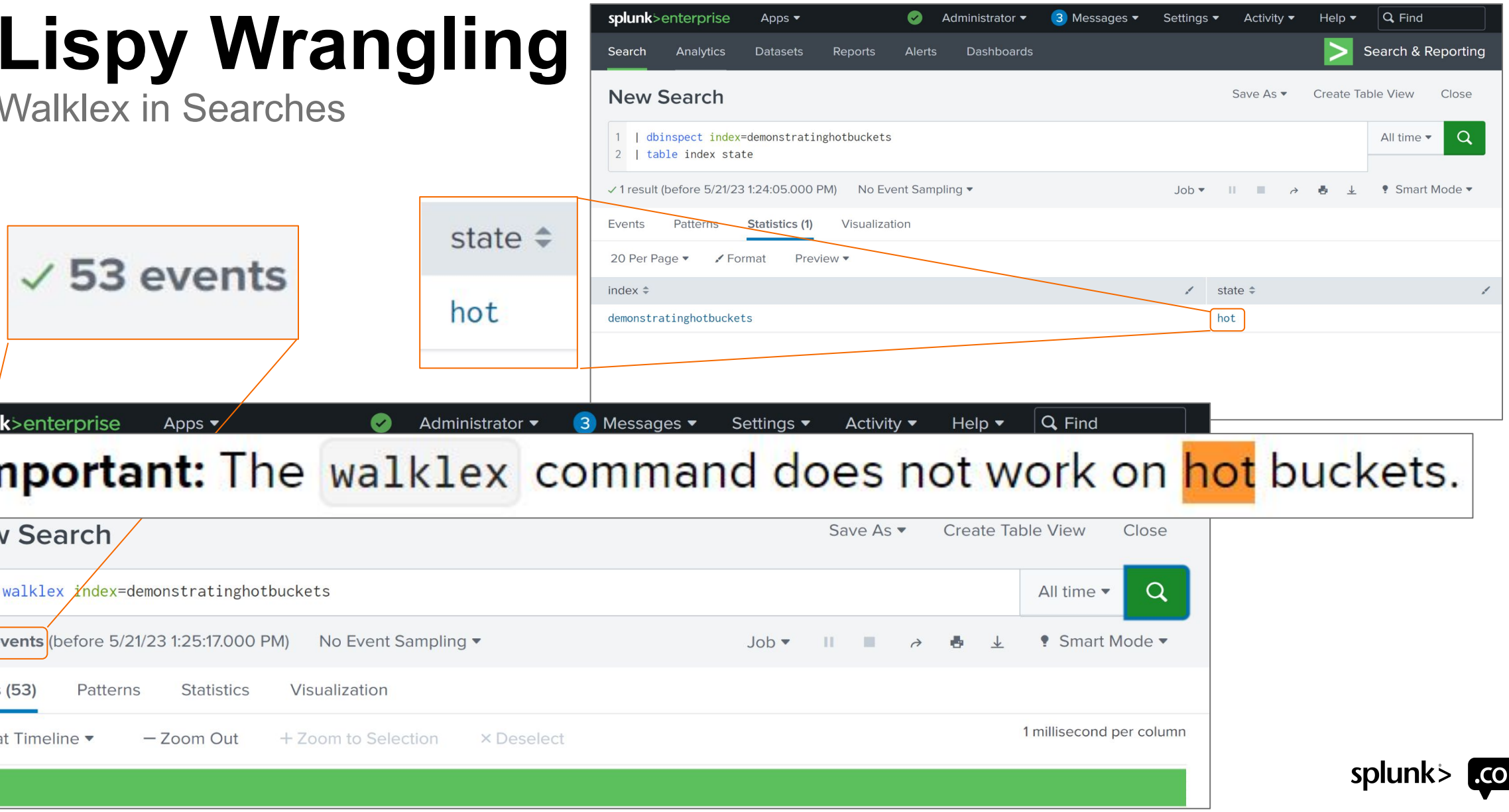
Do you Trust that Pattern?



© 2023 SPLUNK INC.

Lispy Wrangling

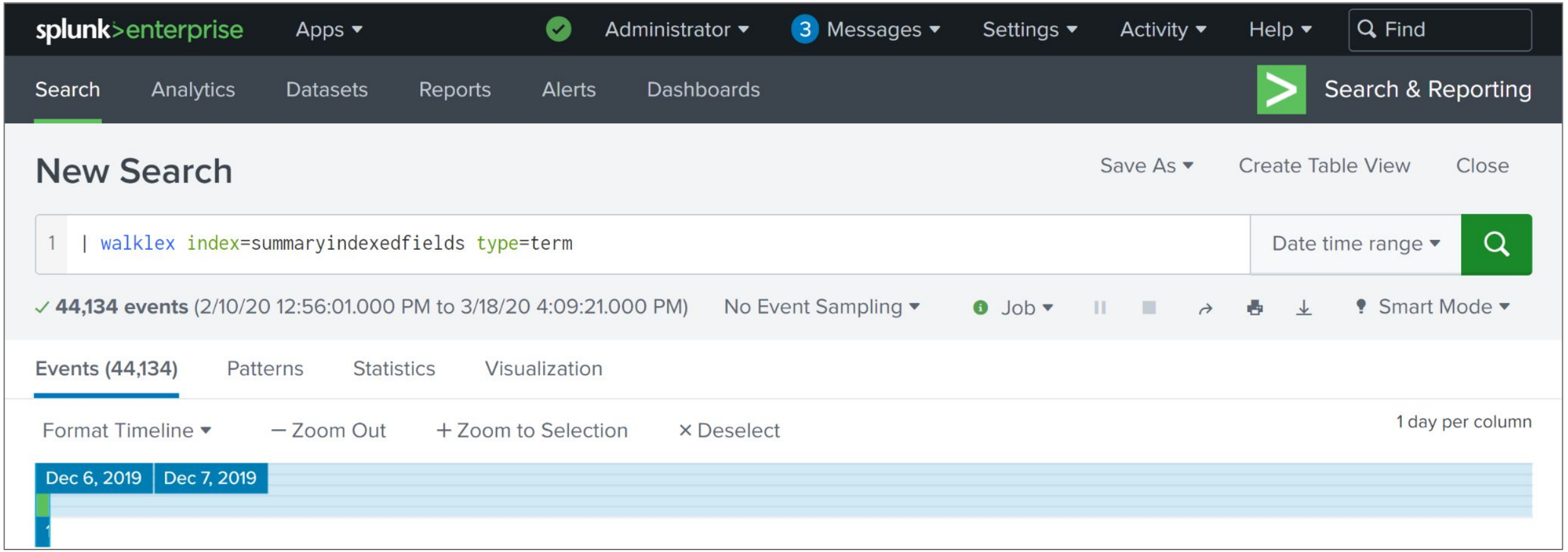
Walklex in Searches



© 2023 SPLUNK INC.

Lispy Wrangling

Walklex in Searches



© 2023 SPLUNK INC.

Lispy Wrangling

Walklex in Searches