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

Splunk, Splunk>, Data-to-Everything, D2E 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. © 2021 Splunk Inc. All rights reserved.



Clara-Fication: More Tstats for Your Buckets

TRU1133B

Clara Merriman Senior Splunk Engineer | Splunk

Martin Müller Principal Consultant | Consist







Martin Müller

Principal Consultant Consist Software Solutions GmbH

Clara Merriman

Senior Splunk Engineer | Splunk



Agenda

| tstats summariesonly=t

values(conf.section)

from datamodel=conf

where conf.talk=TRU1133

by conf.section_number



1) Quick Reintroduction to tstats allow old summaries = true

- 2) Cool Things tstats Can Do TERM(cool=1)
- 3) Converting Searches to tstats
- 4) "Pivot" Job Inspector

5) Optional Arguments append = true





(Re-)Intro to | tstats



Three Types of Stats-y Searches*

Event searches:

index=web
| stats count by host

- infinite flexibility
- speed penalties

Index-based tstats:

| tstats count where index=web by host

limited to index-timevery fast

Datamodel-based tstats:

| tstats count from datamodel=Web by host

- some flexibility at the expense of prep & rebuild efforts
- very fast with DMA

*There's also mstats for metrics indexes, that's a topic for another talk...



Three Types of Stats-y Searches*

Event searches:

index=web
 stats count by host

- infinite flexibility
- speed penalties

Index-based tstats:

| tstats count where index=web by host

- limited to index-time
- very fast

Datamodel-based tstats:

| tstats count from datamodel=Web by host

- some flexibility at the expense of prep & rebuild efforts
- very fast with DMA

*There's also mstats for metrics indexes, that's a topic for another talk...



Three Types of Stats-y Searches*

Event searches:

index=web | stats count by host

- infinite flexibility
- speed penalties

Index-based tstats:

| tstats count where index=web by host

- limited to index-time
- very fast

Datamodel-based tstats:

| tstats count
from datamodel=Web
by host

- some flexibility at the expense of prep & rebuild efforts
- very fast with DMA

*There's also mstats for metrics indexes, that's a topic for another talk...



© 2021 SPLUNK INC.

Cool Examples





• Filter by indexed fields

| tstats count where index=_internal source=*metrics.log*

Filter by simple indexed terms

| tstats count where index=_internal source=*metrics.log* typingqueue

• Filter by complex indexed terms

| tstats count where index=_internal source=*metrics.log* "blocked=true"

- 😥 🚦
- Error in 'TsidxStats': WHERE clause is not an exact query

tstats count where index=_internal source=*metrics.log* TERM(blocked=true)

• Contemporter TERM() forces Splunk to treat blocked=true as one indexed term, ignoring the minor segmenter =



• Filter by indexed fields

| tstats count where index=_internal source=*metrics.log*

Filter by simple indexed terms

| tstats count where index=_internal source=*metrics.log* typingqueue

• Filter by complex indexed terms

| tstats count where index=_internal source=*metrics.log* "blocked=true"

- 😥 🚦
- Error in 'TsidxStats': WHERE clause is not an exact query

tstats count where index=_internal source=*metrics.log* TERM(blocked=true)

• Contemporter TERM() forces Splunk to treat blocked=true as one indexed term, ignoring the minor segmenter =



- Filter by indexed fields
- | tstats count where index=_internal source=*metrics.log*
- Filter by simple indexed terms
- | tstats count where index=_internal source=*metrics.log* typingqueue
- Filter by complex indexed terms

| tstats count where index=_internal source=*metrics.log* "blocked=true"

Error in 'TsidxStats': WHERE clause is not an exact query

tstats count where index=_internal source=*metrics.log* TERM(blocked=true)

• 😅 TERM() forces Splunk to treat blocked=true as one indexed term, ignoring the minor segmenter =



- Filter by indexed fields
- | tstats count where index=_internal source=*metrics.log*
- Filter by simple indexed terms
- | tstats count where index=_internal source=*metrics.log* typingqueue
- Filter by complex indexed terms

| tstats count where index=_internal source=*metrics.log* "blocked=true"

Error in 'TsidxStats': WHERE clause is not an exact query

| tstats count where index=_internal source=*metrics.log* TERM(blocked=true)

• 😅 TERM() forces Splunk to treat blocked=true as one indexed term, ignoring the minor segmenter =



PREFIX() Fun

• This does not work, there is no indexed field name:

| tstats count where index=_internal source=*metrics.log* TERM(blocked=true) by name

PREFIX(name=) will ad-hoc-interpret the indexed term name=typingqueue as a key-value pair (v8.0+)
 tstats count where index=_internal source=*metrics.log* TERM(blocked=true) by PREFIX(name=)

tstats count where index=_internal source=*metrics.log* TERM(g	roup=queue)			Last 24	hours 🔻	
✓ 19,134 events (07/07/2021 16:00:00.000 to 08/07/2021 16:06:45.000)	No Event Sampling 💌	Job 🔻 I	ð 🖪	<u>+</u> ?	Smart Mo	ode 🔻
Events Patterns Statistics (18) Visualization						
20 Per Page 🔻 🖌 Format 🛛 Preview 💌						
name= 🗢	1				COL	unt 🗢 🍃
aeq						1063
aggqueue						1063
aq						1063
auditqueue						1063

PREFIX() Fun

• This does not work, there is no indexed field name:

tstats count where index=_internal source=*metrics.log* TERM(blocked=true) by name

PREFIX(name=) will ad-hoc-interpret the indexed term name=typingqueue as a key-value pair (v8.0+)
 | tstats count where index=_internal source=*metrics.log* TERM(blocked=true) by PREFIX(name=)

tstats count where index=_internal source=*metrics.log* TERM(group=queue) by PREFIX(name=)						Last	24 hours 🔻 🔍
✓ 19,134 events (07/07/2021 16:00:00.000 to 08/07/2021 16:06:45.000) No Event Sampling ▼		Job 🔻	П	ð	•	¥	¶ Smart Mode ▼
Events Patterns Statistics (18) Visualization							
20 Per Page Format Preview							
name= \$	1						count 🗘 🍬
aeq							1063
aggqueue							1063
aq							1063
auditqueue							1063

PREFIX() Magic

• "Traditional" search to chart queue sizes:

Observed 80k Events/s per indexer in a real-world environment... not bad, but can tstats help?

• PREFIX() works for aggregations too!

| tstats max(PREFIX(current_size_kb=)) as max_size_kb where index=_internal source=*metrics.log* TERM(group=queue) by _time span=15m PREFIX(name=) | timechart span=15m max(max_size_kb) by "name="

Same environment returned 2-3M EPS per indexer, 30x speedup

• But remember, your raw event layout matters: name=typingqueue works while name="typingqueue" breaks PREFIX() due to segmentation



PREFIX() Magic

• "Traditional" search to chart queue sizes:

Observed 80k Events/s per indexer in a real-world environment... not bad, but can tstats help?

PREFIX() works for aggregations too!

tstats max(PREFIX(current_size_kb=)) as max_size_kb
where index=_internal source=*metrics.log* TERM(group=queue)
by _time span=15m PREFIX(name=)
timechart span=15m max(max_size_kb) by "name="

Same environment returned 2-3M EPS per indexer, 30x speedup

 But remember, your raw event layout matters: name=typingqueue works while name="typingqueue" breaks PREFIX() due to segmentation



PREFIX() Magic

• "Traditional" search to chart queue sizes:

Observed 80k Events/s per indexer in a real-world environment... not bad, but can tstats help?

```
• PREFIX() works for aggregations too!
```

| tstats max(PREFIX(current_size_kb=)) as max_size_kb where index=_internal source=*metrics.log* TERM(group=queue) by _time span=15m PREFIX(name=) | timechart span=15m max(max_size_kb) by "name="

Same environment returned 2-3M EPS per indexer, 30x speedup

 But remember, your raw event layout matters: name=typingqueue works while name="typingqueue" breaks PREFIX() due to segmentation





From Search to tstats



Basic Conversion

Event Search

index=main
| stats count by sourcetype

tstats Search

| tstats count where index=main by sourcetype



Event Search

tstats Search

index=pan_logs
 stats count

tstats count where index=pan_logs



Event Search

index=pan_logs
| stats count by host

tstats Search

| tstats count where index=pan_logs
by host



Event Search

index=pan_logs timechart span=1d count by host

tstats Search

| tstats prestats=t count where index=pan_logs by _time span=1d host | timechart span=1d count by host



Event Search

index=pan_logs
 eval _time=_indextime
 timechart span=1d count by host

index=pan_logs

| eval _time=_indextime

| timechart span=1d count by host

19,128,155,941 events (6/13/21 12:00:00.000 AM to 7/13/21 8:53:02.000 AM) No Event Sampling *



Search job inspector | Splunk 8.1.2101.2

Search job inspector

This search has completed and has returned 31 results by scanning 19,128,155,941 events in 5,559.225 seconds

(SID: 1626187982.1064426_677D0E40-1944-41AC-AD52-F759224C8AF4) search.log Job Details Dashboard

tstats Search

| tstats prestats=t count where index=pan_logs by _indextime host eval _time=_indextime timechart span=1d count by host

| tstats prestats=t count where index=pan_logs by _indextime host
| eval time= indextime

| timechart span=1d count by host

19,128,188,386 events (6/13/21 12:00:00.000 AM to 7/13/21 8:53:07.000 AM) No Event Sampling *

Search job inspector | Splunk 8.1.2101.2

Search job inspector

🖲 🙆 🥚

This search has completed and has returned 31 results by scanning 19,128,188,386 events in 79.928 seconds

(SID: 1626187987.1064430_677D0E40-1944-41AC-AD52-F759224C8AF4) search.log Job Details Dashboard







Conversion Algorithm

From search to tstats

1) What is the end result?

2) Are the fields needed available at index time?

 | walklex index="<index>" type=field | search NOT field=" *" | stats list(distinct_values) by field

3) Rearrange the search so that the aggregation commands are in the tstats command

4) Use prestats=true when wanting to run a timechart command

5) Add index, source, sourcetype, etc. filters into the WHERE clause



© 2021 SPLUNK INC.

Let's Pivot





Data Models

Job Inspector to the Rescue

Find the | tstats search from a working Pivot using the Job Inspector

Search	Analytics	Datasets	Reports	Alerts	Dashboards						Search & Reporting
_	New P	livot						Save	As 🔻 Cl	ear Edit Data	uset Authentication 💌
	./ 322 523	events (before	7/20/21 5:30	0.30 000 PM						II II O	i a ± ē Q
≁	Filters	events (serore	11201210.00				Split Colum	15			/ Documentation @
-	All time	1	+				ii action			/	/
al	Split Rows						Column Valu	Jes		/	
_	ii abb	-					E Count of	Autn			
=	app \$			/		failure 🗘 🖌		succes	s \$ /		unknown 🗘 🦼
¢	linux_secu	ire				0			0		13
æ	mysql					0			0		36366
÷	pan:system	1				0			0		29
••	ssna					70922			2		e
40	win:local					1			44		(
42	win:remote	2				3339			71505		ę
(7)	win:unknow	'n				195			140100		ę
8											
E											



Data Models

Job Inspector to the Rescue

Note that this only works when the DM is accelerated - pivot on an unaccelerated DM is converted to regular search

Search pay report of spinne 60.4 New Pivot i-066642/16519bb25b7.02.splunkti.0/(en-U5/manager/search/job_inspector?sid=1626827/430.215 iteation Salarch pay report Image: Imag		s Alerts Dashbo	Search School Sc	ch & Reporti
* 322,523 events (before 7/2) Islei None Image: Sector 1 Islei None Image: Sector 1 Islei Sector 1 Secto	New Pivot	i-0664a2fe519bb25b	Search job inspector Splunk 8.0.4 7.o2.splunkit.io/en-US/manager/search/job_inspector?sid=1626827430.215	ntication 🖣
Filters modifiedTime 201-07-2017/3.51.4.77/-07.00 modifiedTime modifiedTime <td>✓ 322.523 events (before 7/20)</td> <td>label</td> <td>None</td> <td></td>	✓ 322.523 events (before 7/20)	label	None	
Interest All time Interest and a state of Authentication * from dataodel-Authentication approx Authentication.app / Authentication.action* * wathentication.action* * wathentication.app * authentication.action* * wathentication.action* * wathentication.action* * appro* * prestats_reserved_** * prevad_** fillual value=NRLL Authentication action* * app* * unterestication* app Interestication* Interestication* Interestication* * app* * unterestication* app Interestication* Interestication* * app* * unterestication* app Interestication* Interesticati	Filtors	modifiedTime	2021-07-20117:35:34.771-07:00	imentatio
iiii app numPreviews None app ‡ numPreviews None iiii app ptimizedSearch I tatas prestate-rue allow_old_sumaries-true count as "Count of Authentication" FROM datamodel=Authentication app" chart None app ‡ iiiiux_secure Previews I tatas prestate-rue allow_old_sumaries-true count as "Count of Authentication.action" = Authentication.action", app='Authentication.app' chart None mysql phase0 tatas prestate-rue allow_old_sumaries-true count as "Count of Authentication" FROM datamodel=Authentication app' chart None sshd phase0 tatas prestate-rue allow_old_sumaries-true count as "Count of Authentication.action" = Authentication.action = Authentication = Authentication.action = Authentication.action = Authentication = Au	All time	normalizedSearch	tstats count AS "Count of Authentication" from datamodel=Authentication where (nodename = Authentication) groupby Authentication.app Authentication.action prestats=true eval "Authentication.action"='Authentication.action', app='Authentication.app' addinfo type=count label=prereport_events fields keepcolorder=t "Authentication.action" "app" "prestats_reserved_=*" "psrsvd_=*" fillnull value=NULL "tothestication action" appetition = location action" "app" "prestats_reserved_=*" psrsvd_=*" fillnull value=NULL	
if opp optimizedSearch tats prestats=true allow_old_summaries=true count as "Count of Authentication" FROM datandel=Authentication app", "Authentication.app", "Authentication.action" wal "Authentication.action" wal "Authentication.action", app='Authentication.app", chart incount app \$ pix		numPreviews	None	
mysql Authentication.app fieldsapan fields + app, **' mysql pan:system tats prestats=true allow_old_sumaries=true count as 'Count of Authentication' FROM datamodel=Authentication app' fillnull pan:system *Authentication.action' eval 'Authentication' app authentication.app' fields + app, '*' authentication.app' fields + app, '*' authentication.action' eval 'Authentication' app' Authentication.action' app authentication.app' fields + app, '*' authentication.app' fields + app, '*' authentication.app' fields + app, '*' authentication.action' eval 'Authentication.action'' app authentication.action' eval 'Authentication.action'' app authentication.action' eval 'Authentication.action'' app authentication.action' eval 'Authentication.action'' app authentication.action'' eval 'Authentication.action'' app auton'' authentication.action'' app authentication action'' app auton'' authentication.action'' app authentication action'' app' authentication'' authentication'' authentication'' app' authentication'' authentication'' app' authentication'' authentication'' authentication'' app	app \$	optimizedSearch	<pre> tstats prestats=true allow_old_summaries=true count as "Count of Authentication" FROM datamodel=Authentication WHERE nodename=Authentication BY "Authentication.app", "Authentication.action" eval "Authentication.action"='Authentication.action", app='Authentication.app' chart dedup_splitvals=t limit=100 useother=t count AS "Count of Authentication" by app Authentication.action format=\$VALS:::EAGGS sort limit=100</pre>	nknown ≎
model phase0 tstats prestats true allow_old_summaries true count as "Count of Authentication" FROM datamodel=Authentication app", "Authentication.app", "Authentication.action" eval "Authentication" app pan:system "Authentication.app", "Authentication.action" eval "Authentication.action" - Authentication.action" app sshd "Authentication.action" eval "Authentication.action" - Authentication.action" app phase1 chart dedup_splitvals t limit=100 useother=t count AS "Count of Authentication.action format=\$VAL\$S:::SAGGS presort 100 auto("Authentication.app") sort limit=100 Authentication.app fields - app, ** noop win:remote pid 30708 win:unknown UI:Pivot reduceSearch sistat dedup_splitvals=t count AS "Count of Authentication.action" reduceSearch sistat dedup_splitvals=t count AS "Count of Authentication.action"	mysal		Authentication.app fieldsspan fields + app, "*"	36
sshd phase1 chart dedup_splitvals=t limit=100 useother=t count AS "Count of Authentication" by app Authentication format=\$VAL5:::\$AGGS presort 100 su-1 auto("Authentication.app") sort limit=100 useother=t count AS "Count of Authentication" by app Authentication format=\$VAL5:::\$AGGS presort 100 win:local pid 30708 win:remote provenance UI:Pivot reduceSearch sistats dedup_splitvals=t count AS "Count of Authentication" by app Authentication.action reduceSearch sistats dedup_splitvals=t count AS "Count of Authentication" FROM datamodel=Authentication WHERE nodename=Authentication BY	pan:system	phase0	tstats prestats=true allow_old_summaries=true count as "Count of Authentication" FROM datamodel=Authentication WHERE nodename=Authentication BY "Authentication.app", "Authentication.action" eval "Authentication.action"='Authentication.action", app='Authentication.app' fillnull value=NULL "Authentication.action" prestats dedup_splitvals=t count by "Authentication.action" app	
win:local pid 30708 win:remote priority 5 win:unknown provenance UI:Pivot reduceSearch sistats dedup_splitvals*t count AS "Count of Authentication" by app Authentication.action ReduceSearch remoteSearch stats prestats*true allow_old_summaries*true count as "Count of Authentication" FROM datamodel~Authentication WERE nodename=Authentication BY	sshd su-1	phase1	chart dedup_splitvals=t limit=100 useother=t count AS "Count of Authentication" by app Authentication.action format=\$VAL\$:::\$AGG\$ presort 100 auto("Authentication.app") sort limit=100 Authentication.app fieldsspan fields + app, "#* noop	
win: remote proviny 5 win: unknown provenance UI: Pivt rduceSearch stats dedup.splitvals*t count AS "Count of Authentication" by app Authentication.action FROM datamodel-Authentication WHERE nodename=Authentication BY	win:local	pid	30708	
win:unknown provenance UI:Pivot reduceSearch sistats dedup_splitvals=t count AS "Count of Authentication" by app Authentication.action reduceSearch remoteSearch tstats prestats=true allow_old_summaries=true count as "Count of Authentication" FROM datamodel=Authentication WHERE nodename=Authentication BY	win:remote	priority	5	
reduceSearch sistats dedup_splitvals=t count AS "Count of Authentication" by app Authentication.action remoteSearch tstats prestats=true allow_old_summaries=true count as "Count of Authentication" FROM datamodel=Authentication WHERE nodename=Authentication BY	win:unknown	provenance	UI:Pivot	
remoteSearch tstats prestats=true allow_old_summaries=true count as "Count of Authentication" FROM datamodel=Authentication WHERE nodename=Authentication BY		reduceSearch	sistats dedup_splitvals=t count AS "Count of Authentication" by app Authentication.action	
"Authentication.app", "Authentication.action" eval "Authentication.action"='Authentication.action", app='Authentication.app' fillnull		remoteSearch	tstats prestats=true allow_old_summaries=true count as "Count of Authentication" FROM datamodel=Authentication WHERE nodename=Authentication BY "Authentication.app", "Authentication.action" eval "Authentication.action"='Authentication.action', app='Authentication.app' fillnull uplumBMEL "Authentication action" eval "Authentication action"='Authentication action', app='Authentication.app' fillnull uplumBMEL "Authentication action" eval "Authentication by "Authentication action" app	



Expanding Pivot

Run pivot in search

Mac: cmd + shift + E Windows: ctrl + shift + E

Pivot will expand into tstats or generic search, depending on DM acceleration

Search Datasets Reports Alerts Dashboards	Expanded Search String
New Search	I totale populational unpricedurables allow all sumprises false owner as "fount of Firewall Large" EDW
pivot pan_firewall log count(log) AS "Count of Firewall Logs" ROWSUMMA	datamodel=pan_firewall WHERE nodename=log
A Your search is paused. No Event Sampling *	fillnull "Count of Firewall Logs"
Events Patterns Statistics (1) Visualization	fields + "Count of Firewall Logs"
100 Per Page 👻 🖍 Format 🛛 Preview 👻	
Count of Firewall Logs \$	Cancel Open as new search

| pivot pan_firewall log count(log) AS "Count
of Firewall Logs" ROWSUMMARY 0 COLSUMMARY 0
SHOWOTHER 1

BECOMES

```
| tstats prestats=true summariesonly=false
allow_old_summaries=false count as "Count of
Firewall Logs" FROM datamodel=pan_firewall
WHERE nodename=log
| stats dedup_splitvals=t count AS "Count of
Firewall Logs"
| fillnull "Count of Firewall Logs"
| fields + "Count of Firewall Logs"
```





Explore Your Options



Datamodel-specific Options

summariesonly=true

- Use accelerated data only, do not fill in the gaps for the last few minutes
- Great for dashboards displaying 24 hours of data
- | tstats summariesonly=true estdc(All_Traffic.src) from datamodel=Network_Traffic by All_Traffic.dvc

allow_old_summaries=true

- Allows tstats to use existing older accelerated data even if the DM changed, e.g. an eval-field was refined
- Favours speed over using the latest search-time knowledge objects
- Most relevant in combination with summariesonly=true, avoids entirely empty results after a change
- | tstats summariesonly=true allow_old_summaries=true count from datamodel=Authentication by _time span=1h



Datamodel-specific Options

summariesonly=true

• Use accelerated data only, do not fill in the gaps for the last few minutes

Great for dashboards displaying 24 hours of data

tstats summariesonly=true estdc(All_Traffic.src) from datamodel=Network_Traffic by All_Traffic.dvc

allow_old_summaries=true

- Allows tstats to use existing older accelerated data even if the DM changed, e.g. an eval-field was refined
- Favours speed over using the latest search-time knowledge objects
- Most relevant in combination with summariesonly=true, avoids entirely empty results after a change

| tstats summariesonly=true allow_old_summaries=true count from datamodel=Authentication by _time span=1h



Good Options for All

fillnull_value="Buttercup"

- Tells tstats what do do when events don't have values for a group-by-field
- Default behaviour: Ignore those events!
- | tstats fillnull_value="Buttercup" count where index=_internal by user

span=1h

- Used when grouping by _time, almost like in the timechart and bin commands
- Aggregates slices of time together by the given span instead of automatically based on time range
- | tstats count from datamodel=Authentication by _time span=1h
- Note: Unlike timechart, tstats will not generate rows for empty spans



Good Options for All

fillnull_value="Buttercup'

• Tells tstats what do do when events don't have values for a group-by-field

Default behaviour: Ignore those events!

tstats fillnull_value="Buttercup" count where index=_internal by user

span=1h

Used when grouping by _time, almost like in the timechart and bin commands

• Aggregates slices of time together by the given span instead of automatically based on time range

| tstats count from datamodel=Authentication by _time span=1h

Note: Unlike timechart, tstats will not generate rows for empty spans



Map-Reduce Shenanigans

prestats=true

- Allows you to specify the Map- and Reduce-parts of tstats independently from each other
- Enables lots of cool hacks coming up in a minute
- | tstats prestats=true count from datamodel=Web by Web.src | 🧙 | stats count by Web.src

append=true

- Lets you concatenate multiple tstats result sets on the indexers, prestats=true only
- Allows 7.x-compatible emulation of fillnull_value without append command:

```
tstats prestats=true count where index=_internal user=* by user
tstats prestats=true append=true count where index=_internal NOT user=*
fillnull value="Buttercup" user
stats count by user
```



Map-Reduce Shenanigans

prestats=true

• Allows you to specify the Map- and Reduce-parts of tstats independently from each other

• Enables lots of cool hacks coming up in a minute

| tstats prestats=true count from datamodel=Web by Web.src | 🧙 | stats count by Web.src

append=true

- Lets you concatenate multiple tstats result sets on the indexers, prestats=true only
- Allows 7.x-compatible emulation of fillnull_value without append command:

```
tstats prestats=true count where index=_internal user=* by user
tstats prestats=true append=true count where index=_internal NOT user=*
fillnull value="Buttercup" user
stats count by user
```



prestats=true Basics

• Every search has a Map- and a Reduce-phase

- Job inspector: phase0 & phase1
- Event search phase0 is "everything including the first stats", phase1 is "everything from the first stats"
- tstats is its own generating command and stats combined, therefore tstats is split up automatically: search: | tstats count where index= internal by sourcetype

phase0: | tstats prestats=t <snip> count where index=_internal groupby sourcetype

phase1: | tstats <snip> count WHERE index=_internal BY sourcetype

• That's equivalent to specifying both phases manually:

tstats prestats=t count where index=_internal by sourcetype
stats count by sourcetype



prestats=true Basics

- Every search has a Map- and a Reduce-phase
- Job inspector: phase0 & phase1
- Event search phase0 is "everything including the first stats", phase1 is "everything from the first stats"
- tstats is its own generating command and stats combined, therefore tstats is split up automatically:
- search: | tstats count where index=_internal by sourcetype
- phase0: | tstats prestats=t <snip> count where index=_internal groupby sourcetype
- phase1: | tstats <snip> count WHERE index=_internal BY sourcetype
- That's equivalent to specifying both phases manually:
 - tstats prestats=t count where index=_internal by sourcetype
 stats count by sourcetype



prestats=true Under the Hood

• What if you "forget" the manual phase1 / Reduce?

tstats	prestats=t	count where	index=_internal by sourcety	pe
✓ 16 events	s (04/07/202	1 16:00:00.00	0 to 05/07/2021 16:20:16.000)	No Event Sampling 🔻
Events	Patterns	Statistics	Visualization	
List 🔹 🥻	 Format 	20 Per Page	e 🔻	
i Time	Event			
>	sourcetyp	e = splunk_w	eb_access	
>	sourcetyp	e = splunk_w	eb_service	
>	sourcetyp	e = splunkd		



prestats=true Under the Hood

• What if you "forget" the manual phase1 / Reduce?

Event A	Actio	ns 🔻		
Туре	~	Field	Value	Actions
Selected		sourcetype	splunkd	~
Event		psrsvd_gc	10203	~
		psrsvd_v	1	~

- psrsvd_gc = "prestats-reserved global count", a partial contribution to the total count for splunkd
- You can do SPL with those fields! On the Indexers!
- Touching group-by fields (sourcetype) is usually safe, don't touch aggregations (count / psrsvd_gc)



prestats=true Example: Lookups

• Let the Searchhead do the lookup work all on its own

Let the Indexers share the lookup work between themselves in parallel

tstats prestats=t sum(All_Traffic.bytes) from datamodel=Network_Traffic by All_Traffic.src_ip
lookup dnslookup clientip as "All_Traffic.src_ip" OUTPUT clienthost
search clienthost="*.splunk.com"
stats sum(All_Traffic.bytes) as splunk_bytes
Searchhead

• Note: Renames with prestats=t are ignored, the reduce-stats has to refer to the original name



prestats=true Example: Naïve Approach

Enterprise Security Content Update: SQL Injection with Long URLs

Load all URLs from the Web datamodel to the searchhead, apply complex filtering

tstats summariesonly=t count from datamodel=Web by Web.url
eval num_sql_cmds = <complex eval to count SQL-y looking things>
where num_sql_cmds > 3

This search has completed and has returned 22 results by scanning 536,216,726 events in 38.779 seconds

51.42	dispatch.stream.remote	1,182	-	1,081,017,538

• Over 1GB of data returned to SH, 2.5M rows need to be merged and sorted before eval

Single SH doing filtering



prestats=true Example: Smart Approach

Enterprise Security Content Update: SQL Injection with Long URLs

• Offload the complex filtering to the indexers, only load matching URLs to the searchhead

tstats summariesonly=t prestats=t count from datamodel=Web by Web.url
eval num_sql_cmds = <complex eval to count SQL-y looking things>
where num_sql_cmds > 3
stats count by Web.url

This search has completed and has returned 22 results by scanning 536,216,726 events in 6.881 seconds

60.88	dispatch.stream.remote	1,182	~	9,072,147

- Under 10MB of data returned to SH
- Parallel filtering on every indexer
- 6x speedup



Key Takeaways





What to Remember

- tstats works on indexed fields and datamodels
- ctrl+shift+E or cmd+shift+E will expand the pivot search and show tstats
- prestats=t is great when additional aggregations follow
- FROM specifies the datamodel
- WHERE filters events
- PREFIX() and TERM() are magic
- Be mindful of high cardinality intermediate results



© 2021 SPLUNK INC.

Resources





Helpful Links



- More on the Job Inspector: <u>https://conf.splunk.com/watch/conf-online.html?searc</u> <u>h=TRU1143C</u>
- tstats Reference Docs: <u>https://docs.splunk.com/Documentation/Splunk/latest/</u> <u>SearchReference/Tstats</u>
- pivot Reference Docs: <u>https://docs.splunk.com/Documentation/Splunk/latest/</u> <u>SearchReference/Pivot</u>
- Additional Background on Map-Reduce: <u>https://en.wikipedia.org/wiki/MapReduce</u>



• -

 \bigcirc

Thank You

Please provide feedback via the

SESSION SURVEY

