

Power Of Splunk SPL (Search Processing Language)

Stephen Luedtke

Sr. Technical Marketing Mgr, Splunk

.conf2016

splunk >

Disclaimer

During the course of this presentation, we may make forward looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC. The forward-looking statements made in the this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not, be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

Agenda

- Overview & Anatomy of a Search
 - Quick refresher on search language and structure
- SPL Commands and Examples
 - Searching, *charting*, converging, mapping, transactions, anomalies, exploring
- Custom Commands
 - Extend the capabilities of SPL
- Q&A

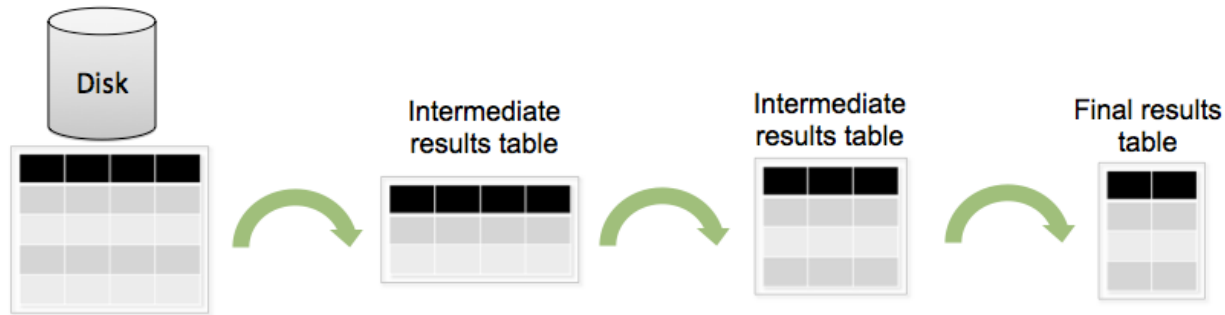
SPL Overview

.conf2016

splunk >

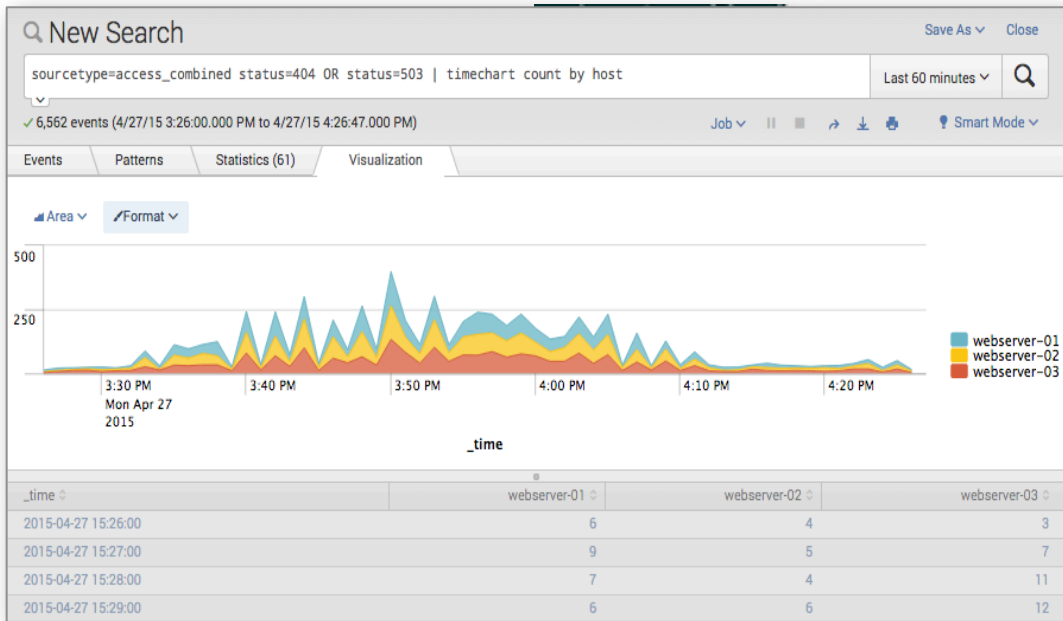
SPL Overview

- Over 140+ search commands
- Syntax was originally based upon the [Unix pipeline](#) and [SQL](#) and is optimized [for time series data](#)
- The scope of SPL includes data searching, filtering, modification, manipulation, enrichment, insertion and deletion
- Includes anomaly detection and machine learning



Why Create A New Query Language?

- Flexibility and effectiveness on *small* and *big* data
- Late-binding schema
- More/better methods of correlation
- Not just analyze, but visualize



SPL Basic Structure

search and filter | **munge** | **report** | cleanup

```
sourcetype=access*
```

```
| eval KB=bytes/1024
```

```
| stats sum(KB) dc(clientip)
```

```
| rename sum(KB) AS "Total KB" dc(clientip) AS "Unique Customers"
```

SPL Examples



.conf2016

SPL Examples And Recipes

- Find the needle in the haystack
- Charting statistics and predicting values
- Enriching and converging data sources
- Visualize geographic data in real time
- Identifying transactions and anomalies
- Data exploration & finding relationships between fields

SPL Examples And Recipes

- **Find the needle in the haystack**
- Charting statistics and predicting values
- Enriching and converging data sources
- Visualize geographic data in real time
- Identifying transactions and anomalies
- Data exploration & finding relationships between fields

Search And Filter

Examples

- **Keyword search:**
`sourcetype=access* http`
- **Filter:**
`sourcetype=access* http`
`host=webserver-02`
- **Combined:**
`sourcetype=access* http`
`host=webserver-02 (503 OR 504)`

The screenshot shows a Splunk search interface with the query `sourcetype=access* http` entered in the search bar. The results are displayed in a table with two columns: 'Matching terms' and 'How to Search'. The 'Matching terms' column lists various URLs and their counts, such as `120,976 http`, `2,154 http://m.acme.com/`, and `4,308 http://m.acme.com/search.php`. The 'How to Search' column provides instructions on how to refine the search, including 'Step 1: Retrieve Events' and 'Step 2: Use Search Commands'. A red circle highlights the 'Matching terms' column header.

Matching terms	How to Search
120,976 http	Step 1: Retrieve Events The simplest searches return events that match terms you type into the search bar. terms: error login quoted phrases: "database error" boolean operators: login NOT (error OR fail) wildcards: fail* field values: status=404, status!=404, or status>200
2,154 http://m.acme.com/	
4,308 http://m.acme.com/search.php	
50,184 http://shop.acme.com/cart.do	
20,055 http://shop.acme.com/category.screen	
13,808 http://shop.acme.com/oldlink	
28,299 http://shop.acme.com/product.screen	
798 http://www.google.com/bot.html	
517 *http_ops=23*	Step 2: Use Search Commands More advanced searches use commands to transform, filter, and report on the events you retrieved. Use the vertical bar " " , or pipe character, to apply a command to the retrieved events.
554 *http_ops=28*	
517 *http_ops_2=23*	
554 *http_ops_2=28*	
557 httpjspbase.java:87	
15,727 https	
557 httpservlet.java:856	

Search And Filter

Examples

- Keyword search:
sourcetype=access* http
- **Filter:**
sourcetype=access* http
host=webserver-02
- **Combined:**
sourcetype=access* http
host=webserver-02 (503 OR 504)

The screenshot shows the Splunk search interface. The search bar contains the query: `sourcetype=access_combined host=web*`. The `host=web*` portion is circled in red. Below the search bar, it indicates 8,839 events were found for the search on 4/7/15 between 8:49:00 AM and 9:49:08 AM. A histogram shows the distribution of events over time. The event list below shows several results, including one with a JavaScript alert triggered from a search page.

i	Time	Event
>	4/7/15 9:49:07.088 AM	175.45.177.187 - - [07/Apr/2015 09:49:07:088873] "POST /search.php?uid=1 AND (SELECT 1 FROM (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'MYSQL_USER')) AS A" 503 45052 "http://m.acme.com/search.php?uid=1 AND (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'MYSQL_USER')) AS A" 6 (KHTML, like Gecko) Version/5.1 Mobile/9A334 Safari/7534.48.3" 8038 host = webserver-03 ; source = /opt/apache/log/access_combined.log ; sourcetype = access_combined
>	4/7/15 9:49:07.070 AM	175.45.177.13 - - [07/Apr/2015 09:49:07:070283] "POST /search.php?uid=01226056-999 "http://m.acme.com/<script type='text/javascript'>alert('test');</script>" AppleWebKit (KHTML, like Gecko) Mobile [FBAN/FBForiPhone;FBAV/4.0.3;FBBV/4030.0;FBDVT;FBLC/en_US;FBSF/1.0]" 11725 host = webserver-03 ; source = /opt/apache/log/access_combined.log ; sourcetype = access_combined
>	4/7/15 9:49:07.067 AM	175.45.177.188 - - [07/Apr/2015 09:49:07:067717] "POST /search.php?uid=b066eae31978 "http://m.acme.com/<script type='text/javascript'>alert('pwd');</script>" AppleWebKit/528.18 (KHTML, like Gecko) Mobile/7E18" 9475 host = webserver-02 ; source = /opt/apache/log/access_combined.log ; sourcetype = access_combined
>	4/7/15 9:49:07.046 AM	175.45.177.13 - - [07/Apr/2015 09:49:07:046441] "POST /search.php?uid=9b04682d2d&JSESSIONID=SD35L1FF7ADFF9 HTTP 1.1" 503 48015 "http://m.acme.com/search.php?uid=9b04682d2d&JSESSIONID=SD35L1FF7ADFF9 HTTP 1.1" (iPhone; CPU iPhone OS 5_0 like Mac OS X) AppleWebKit/534.46 (KHTML, like Gecko) Version/5.0.1 Mobile/9A334 Safari/7534.48.3" 8038 host = webserver-02 ; source = /opt/apache/log/access_combined.log ; sourcetype = access_combined

Search And Filter

Examples

- **Keyword search:**
`sourcetype=access* http`
- **Filter:**
`sourcetype=access* http`
`host=webserver-02`
- **Combined:**
`sourcetype=access* http`
`host=webserver-02 (503 OR 504)`

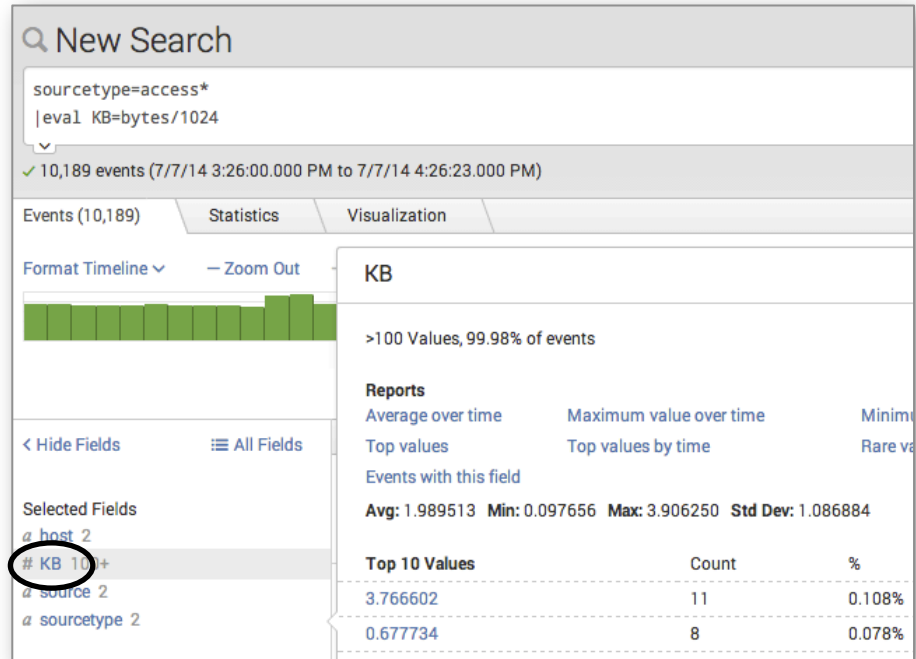
The screenshot shows the Splunk search interface. At the top, the search bar contains the query: `sourcetype=access* host=webserver-02 (503 OR 504)`. Below the search bar, it indicates that 1,965 events were found for the time range 4/7/15 8:53:00.000 AM to 4/7/15 9:53:06.000 AM. The interface includes tabs for Events (1,965), Patterns, Statistics, and Visualization. A timeline visualization shows the distribution of events over time. Below the timeline, there is a table of search results with columns for Time and Event. The table shows four events, each with a timestamp and a detailed log entry. The log entries include IP addresses, user agents, and request details.

i	Time	Event
>	4/7/15 9:52:52.018 AM	144.185.205.147 - - [07/Apr/2015 09:52:52:018196] "GET /oldlink?item_id=W PSS-2&JS duct.screen?product_id=W PSS-2" "mozilla/5.0 (iPad; U; CPU iPhone OS 5_0_1 like Ma one; FBAW/4.0.3; FBBV/4030.0; FBDV/iPad2,1; FBMD/iPad; FBSN/iPhone OS; FBSV/5.0.1; FB5S/ host = webserver-02 ; source = /opt/apache/log/access_combined.log ; sourcetype = access_combined
>	4/7/15 9:52:51.200 AM	175.45.177.187 - - [07/Apr/2015 09:52:51:200299] "POST /search.php?1 AND (SELECT 1FF7ADFF9 HTTP 1.1" 504 53332 "http://m.acme.com/search.php?1 AND (SELECT * FROM 22) AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1" 9877 host = webserver-02 ; source = /opt/apache/log/access_combined.log ; sourcetype = access_combined
>	4/7/15 9:52:51.172 AM	175.45.177.17 - - [07/Apr/2015 09:52:51:172965] "POST /search.php?1 AND (SELECT FF7ADFF9 HTTP 1.1" 503 48362 "http://m.acme.com/search.php?1 AND (SELECT * FROM ppleWebKit/531.21.10 (KHTML, like Gecko) Version/4.0.4 Mobile/7B314 Safari/531.21 host = webserver-02 ; source = /opt/apache/log/access_combined.log ; sourcetype = access_combined
>	4/7/15 9:52:51.013 AM	175.45.177.13 - - [07/Apr/2015 09:52:51:013986] "POST /search.php?&uid=497625e1-6 1648 "http://m.acme.com/<script type='text/javascript'>alert('pwnd');</script>" " ppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1" 10071 host = webserver-02 ; source = /opt/apache/log/access_combined.log ; sourcetype = access_combined

Eval – Modify Or Create New Fields And Values

Examples

- **Calculation:**
`sourcetype=access*`
`| eval KB=bytes/1024`
- **Evaluation:**
`sourcetype=access*`
`| eval http_response =`
`if(status != 200, "Error", "OK")`
- **Concatenation:**
`sourcetype=access*`
`| eval connection = clientip.":".port`



Eval – Modify Or Create New Fields And Values

Examples

- **Calculation:**
`sourcetype=access*`
`| eval KB=bytes/1024`
- **Evaluation:**
`sourcetype=access*`
`| eval http_response =`
`if(status != 200, "Error", "OK")`
- **Concatenation:**
`sourcetype=access*`
`| eval connection = clientip.":".port`

New Search

```
sourcetype=access*
| eval http_response = if(status == 200, "OK", "Error")
```

10,323 events (6/27/14 11:58:00.000 AM to 6/27/14 12:58:58.000 PM)

Events (10,323) | Statistics | Visualization

Format Timeline | Zoom Out | Zoom to Selection | Deselect

List | Format | 20 Per Page

< Hide Fields | All Fields

Selected Fields

- # bytes 100+
- a clientip 100+
- a host 2
- a http_response 2**
- # status 10
- a status_description 40

Interesting Fields

- a action 5
- a bc_uri 100+
- a category_id 9

http_response

2 Values, 100% of events

Selected

Reports

- Top values
- Top values by time
- Rare values

Events with this field

Values	Count	%
OK	7,013	67.936%
Error	3,310	32.064%

12:58:53.562 PM LE-5WL&JSESSIONID=5D5SL1FF1ADFF9 HTTP 1.1" 200 474 "http://shop.splunk.com/..."

Eval – Modify Or Create New Fields And Values

Examples

- **Calculation:**
`sourcetype=access*`
`| eval KB=bytes/1024`
- **Evaluation:**
`sourcetype=access*`
`| eval http_response =`
`if(status != 200, "Error", "OK")`
- **Concatenation:**
`sourcetype=access*`
`| eval connection = clientip.":".port`

New Search

```
sourcetype=access*
| eval connection = clientip.".\".port
```

10,330 events (6/27/14 12:03:00.000 PM to 6/27/14 1:03:52.000 PM)

Events (10,330) | Statistics | Visualization

Format Timeline | Zoom Out | Zoom to Selection | Deselect

connection

>100 Values, 26.941% of events

Selected Yes No

Reports

- Top values
- Top values by time
- Rare values
- Events with this field

Top 10 Values	Count	%
10.120.133.110:80	17	0.611%
10.122.183.49:80	13	0.467%
10.187.165.92:80	13	0.467%
10.169.199.125:80	11	0.395%

Selected Fields

- # bytes 100+
- a clientip 100+
- a connection 100+
- a host 2
- # status 40
- a status_description 40

Eval – Just Getting Started!

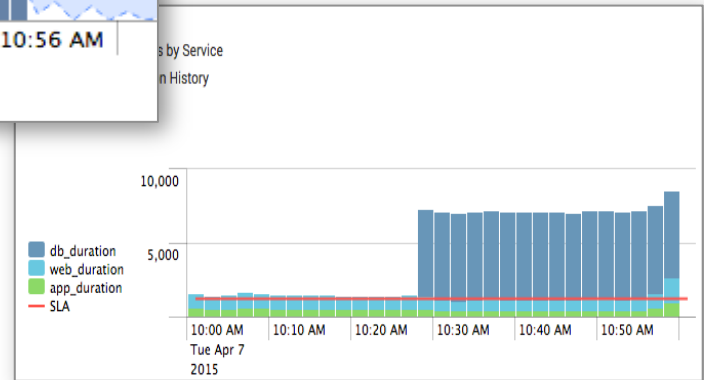
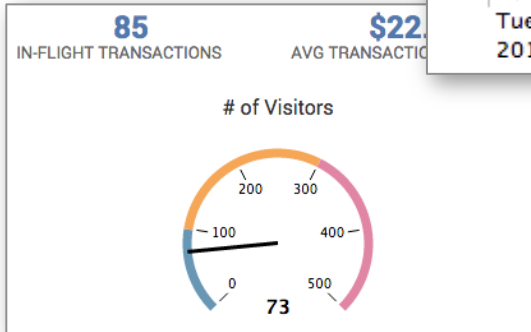
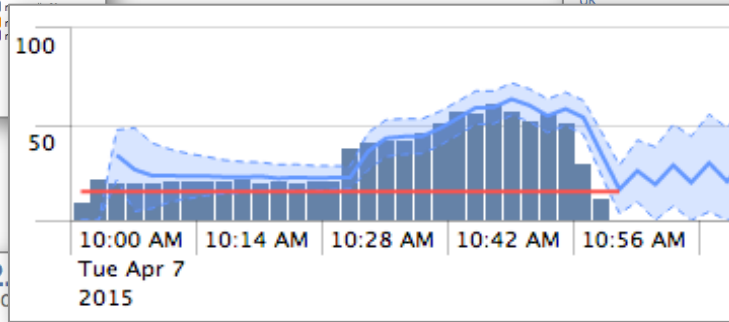
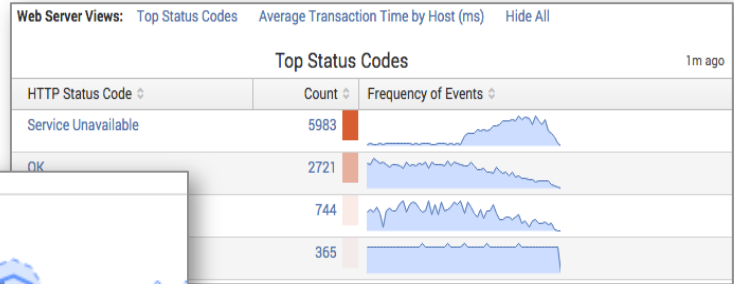
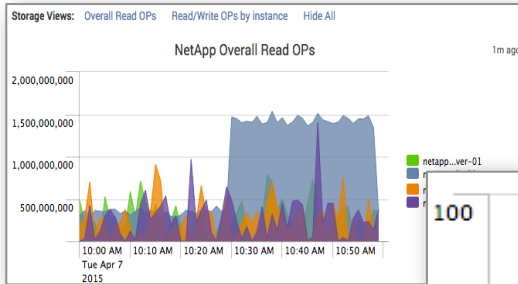
Splunk Search Quick Reference Guide

EVAL FUNCTIONS		
<small>The eval command calculates an expression and puts the resulting value into a field (e.g. "... eval force = mass * acceleration"). The following table lists the functions eval understands, in addition to basic arithmetic operators (+ - * / %), string concatenation (e.g. "... eval name = last .", ". last"), boolean operations (AND OR NOT XOR < > <= >= != == LIKE).</small>		
FUNCTION	DESCRIPTION	EXAMPLES
abs (X)	Returns the absolute value of X.	abs (number)
case (X, "Y", ...)	Takes pairs of arguments X and Y, where X arguments are Boolean expressions that, when evaluated to TRUE, return the corresponding Y argument.	case(error == 404, "Not found", error == 500, "Internal Server Error", error == 200, "OK")
ceil (X)	Ceiling of a number X.	ceil (1.9)
cidrmatch ("X", Y)	Identifies IP addresses that belong to a particular subnet.	cidrmatch ("123.132.32.0/25", ip)
coalesce (X, ...)	Returns the first value that is not null.	coalesce(null(), "Returned val", null())
exact (X)	Evaluates an expression X using double precision floating point arithmetic.	exact (3.14*num)
exp (X)	Returns e ^X .	exp (3)
floor (X)	Returns the floor of a number X.	floor (1.9)
if (X, Y, Z)	If X evaluates to TRUE, the result is the second argument Y. If X evaluates to FALSE, the result evaluates to the third argument Z.	if(error==200, "OK", "Error")
isbool (X)	Returns TRUE if X is Boolean.	isbool (field)
isint (X)	Returns TRUE if X is an integer.	isint (field)
isnotnull (X)	Returns TRUE if X is not NULL.	isnotnull (field)
isnull (X)	Returns TRUE if X is NULL.	isnull (field)
isnum (X)	Returns TRUE if X is a number.	isnum (field)
isstr ()	Returns TRUE if X is a string.	isstr (field)

SPL Examples And Recipes

- Find the needle in the haystack
- **Charting statistics and predicting values**
- Enriching and converging data sources
- Visualize geographic data in real time
- Identifying transactions and anomalies
- Data exploration & finding relationships between fields

Stats, Chart, Timechart



Shake!!

Go to splunk.com/shake
on your mobile device.

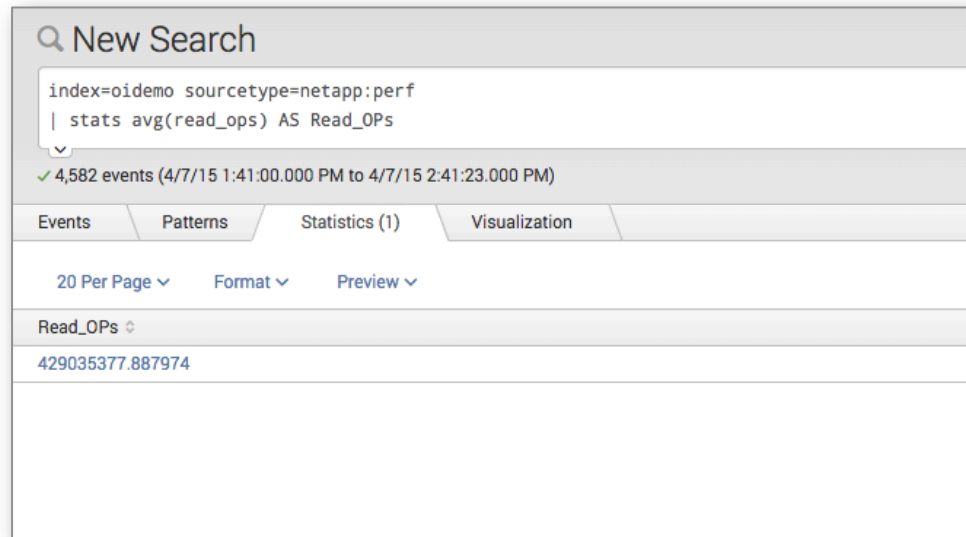
.conf2016

splunk >

Stats-Calculate Statistics Based On Field Values

Examples

- **Calculate stats and rename**
sourcetype=netapp:perf
| stats avg(read_ops) AS "Read OPs"
- **Multiple statistics**
sourcetype=netapp:perf
| stats avg(read_ops) AS Read_OPs
sparkline(avg(read_ops)) AS Read_Trend
- **By another field**
Sourcetype=netapp:perf
| stats avg(read_ops) AS Read_OPs
sparkline(avg(read_ops)) AS Read_Trend
by instance



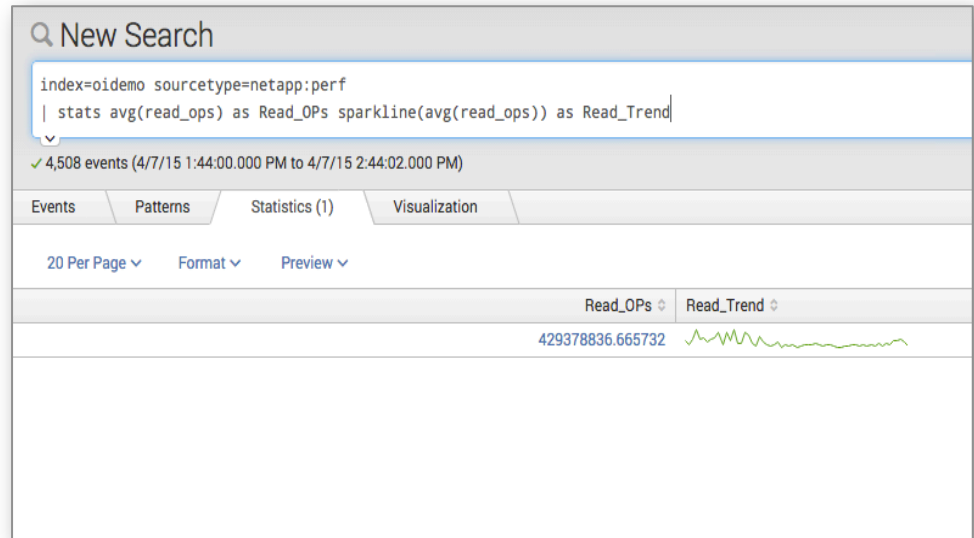
The screenshot shows a Splunk search interface. At the top, there is a search bar with the text "New Search". Below the search bar, the search query is displayed: `index=oidemo sourcetype=netapp:perf | stats avg(read_ops) AS Read_OPs`. The search results show a success message: "4,582 events (4/7/15 1:41:00.000 PM to 4/7/15 2:41:23.000 PM)". Below this, there are tabs for "Events", "Patterns", "Statistics (1)", and "Visualization". The "Statistics (1)" tab is selected, and it shows a table with one row of data. The table has a header row with "Read_OPs" and a data row with the value "429035377.887974".

Read_OPs
429035377.887974

Stats-Calculate Statistics Based On Field Values

Examples

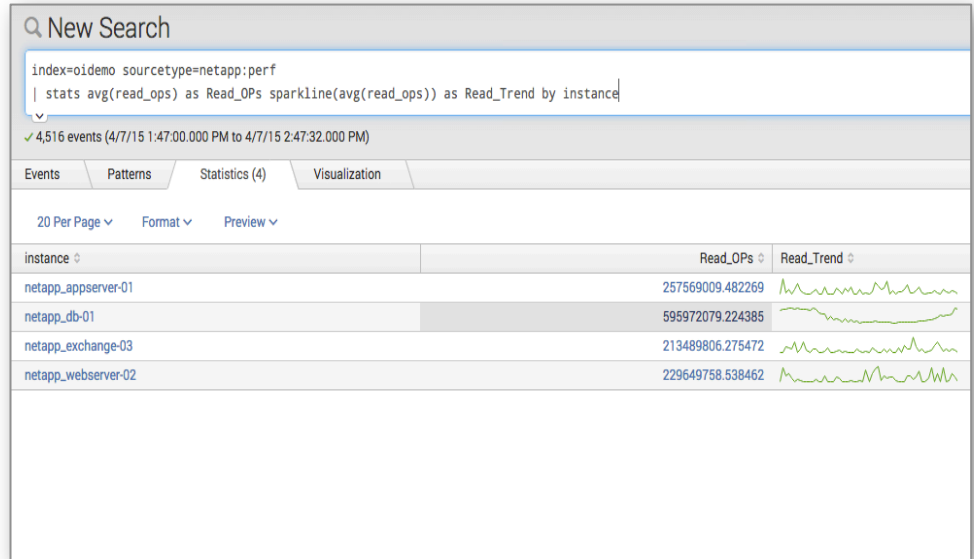
- Calculate stats and rename
sourcetype=netapp:perf
| stats avg(read_ops) AS "Read OPs"
- **Multiple statistics**
sourcetype=netapp:perf
| stats avg(read_ops) AS Read_OPs
sparkline(avg(read_ops)) AS Read_Trend
- By another field
Sourcetype=netapp:perf
| stats avg(read_ops) AS Read_OPs
sparkline(avg(read_ops)) AS Read_Trend
by instance



Stats-Calculate Statistics Based On Field Values

Examples

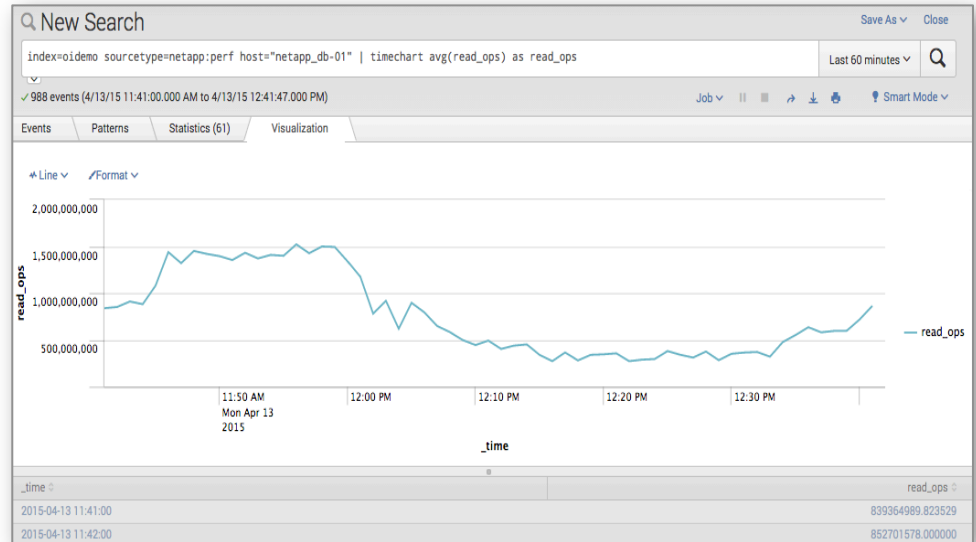
- Calculate stats and rename
sourcetype=netapp:perf
| stats avg(read_ops) AS "Read OPs"
- Multiple statistics
sourcetype=netapp:perf
| stats avg(read_ops) AS Read_OPs
sparkline(avg(read_ops)) AS Read_Trend
- **By another field**
Sourcetype=netapp:perf
| stats avg(read_ops) AS Read_OPs
sparkline(avg(read_ops)) AS
Read_Trend by instance



Timechart – Visualize Statistics Over Time

Examples

- **Visualize stats over time**
sourcetype=netapp:perf
| timechart avg(read_ops)
- **Add a trendline**
sourcetype=netapp:perf
| timechart avg(read_ops) as
read_ops | trendline sma5(read_ops)
- **Add a prediction overlay**
sourcetype=netapp:perf
| timechart avg(read_ops) as
read_ops | predict read_ops



Timechart – Visualize Statistics Over Time

Examples

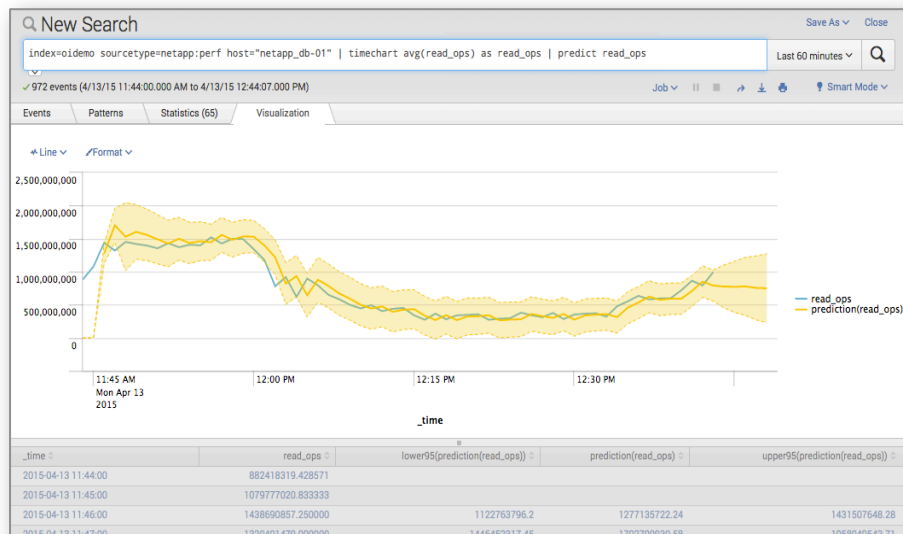
- Visualize stats over time
`sourcetype=netapp:perf`
`| timechart avg(read_ops)`
- **Add a trendline**
`sourcetype=netapp:perf`
`| timechart avg(read_ops) as read_ops | trendline sma5(read_ops)`
- **Add a prediction overlay**
`sourcetype=netapp:perf`
`| timechart avg(read_ops) as read_ops | predict read_ops`



Timechart – Visualize Statistics Over Time

Examples

- Visualize stats over time
`sourcetype=netapp:perf`
`| timechart avg(read_ops)`
- Add a trendline
`sourcetype=netapp:perf`
`| timechart avg(read_ops) as read_ops`
`read_ops | trendline sma5(read_ops)`
- Add a prediction overlay
`sourcetype=netapp:perf`
`| timechart avg(read_ops) as read_ops`
`read_ops | predict read_ops`



Stats/Timechart – But Wait, There's More!

Splunk Search Quick Reference Guide

COMMON STATS FUNCTIONS	
FUNCTION	DESCRIPTION
avg (X)	Returns the average of the values of field X.
count (X)	Returns the number of occurrences of the field X. To indicate a specific field value to match, format X as eval(field="value").
dc (X)	Returns the count of distinct values of the field X.
first (X)	Returns the first seen value of the field X. In general, the first seen value of the field is the chronologically most recent instance of field.
last (X)	Returns the last seen value of the field X.
list (X)	Returns the list of all values of the field X as a multi-value entry. The order of the values reflects the order of input events.
max (X)	Returns the maximum value of the field X. If the values of X are non-numeric, the max is found from lexicographic ordering.
median (X)	Returns the middle-most value of the field X.
min (X)	Returns the minimum value of the field X. If the values of X are non-numeric, the min is found from lexicographic ordering.
mode (X)	Returns the most frequent value of the field X.
perc<X> (Y)	Returns the X-th percentile value of the field Y. For example, perc5(total) returns the 5th percentile value of a field "total".
range (X)	Returns the difference between the max and min values of the field X.
stdev (X)	Returns the sample standard deviation of the field X.
stdevp (X)	Returns the population standard deviation of the field X.
sum (X)	Returns the sum of the values of the field X.
sumsq (X)	Returns the sum of the squares of the values of the field X.
values (X)	Returns the list of all distinct values of the field X as a multi-value entry. The order of the values is lexicographical.
var (X)	Returns the sample variance of the field X.

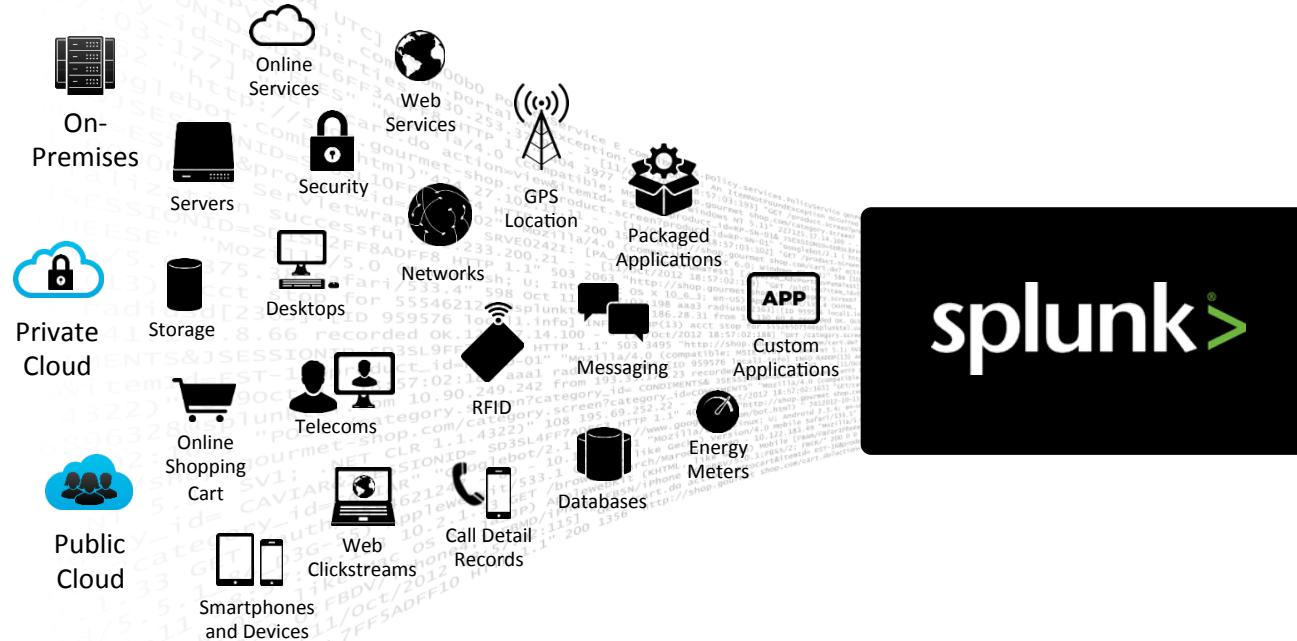
Common statistical functions used with the chart, stats, and timechart commands. Field names can be wildcarded, so avg(*delay) might calculate the average of the delay and xdelay fields.

SPL Examples And Recipes

- Search and filter + creating/modifying fields
- Charting statistics and predicting values
- **Enriching and converging data sources**
- Visualize geographic data in real time
- Identifying transactions and anomalies
- Data exploration & finding relationships between fields

Converging Data Sources

Index Untapped Data: Any Source, Type, Volume



Ask Any Question

Application Delivery

IT Operations

Security, Compliance and Fraud

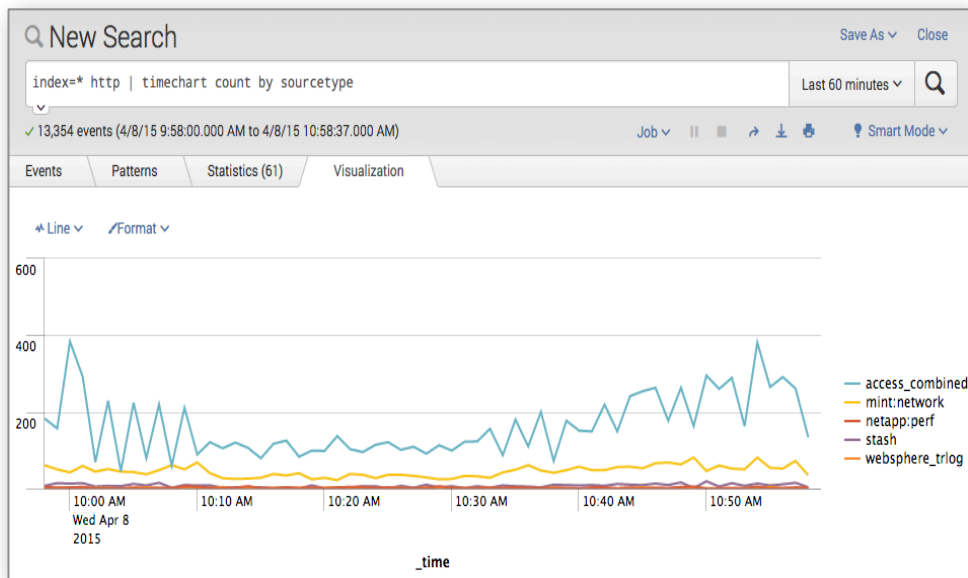
Business Analytics

Industrial Data and the Internet of Things

Converging Data Sources

Examples

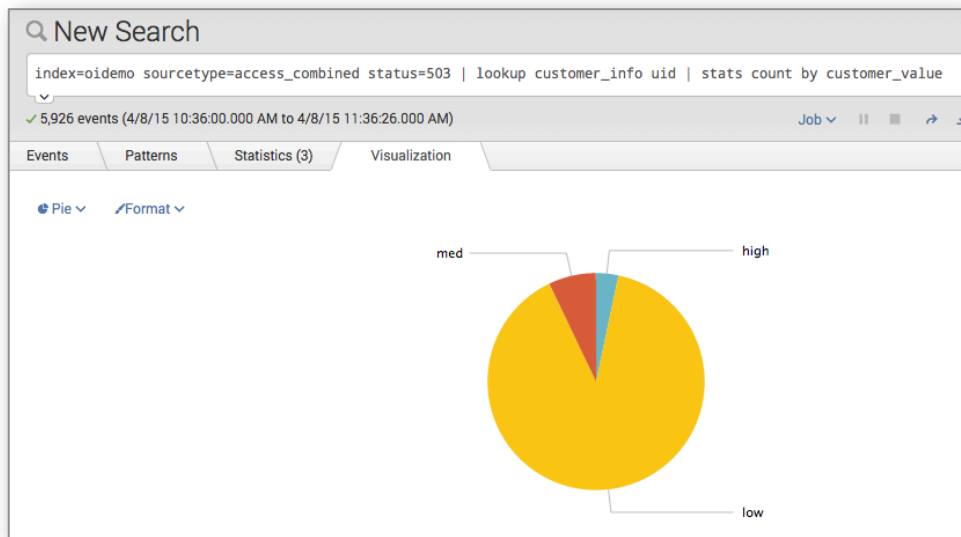
- **Implicit join on time**
`index=* http | timechart count by sourcetype`
- **Enrich data with lookup**
`sourcetype=access_combined status=503
| lookup customer_info uid |
stats count by customer_value`
- **Append results from another search**
`... | appendcols [search earliest=-1h
sourcetype=Kepware units=W row=A
| stats stdev(Value) as hr_stdev] ...`



Lookup – Converging Data Sources

Examples

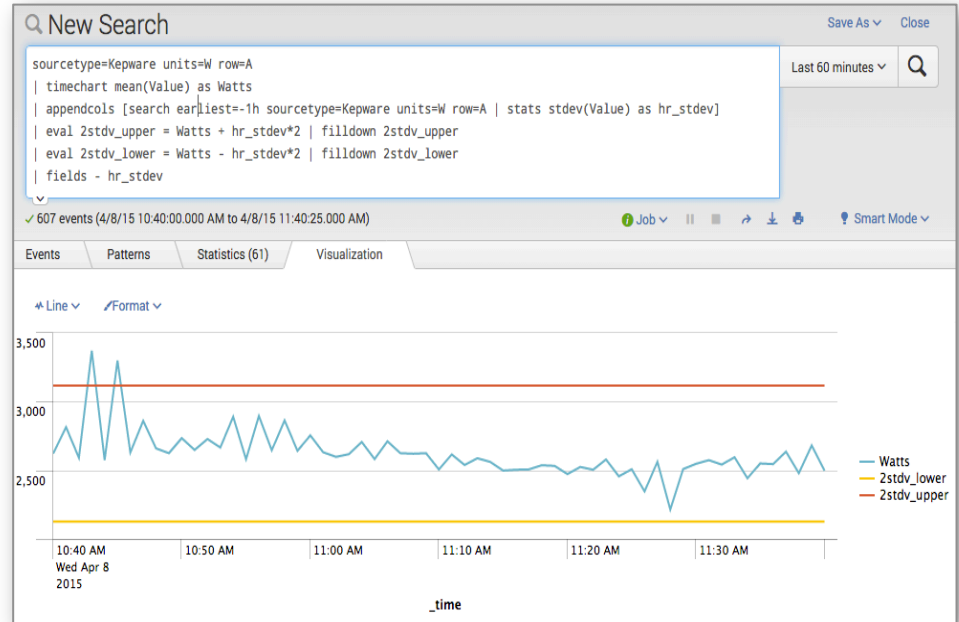
- **Implicit join on time**
`index=* http | timechart count by sourcetype`
- **Enrich data with lookup**
`sourcetype=access_combined status=503 | lookup customer_info uid | stats count by customer_value`
- **Append results from another search**
`... | appendcols [search earliest=-1h sourcetype=Kepware units=W row=A | stats stdev(Value) as hr_stdev] ...`



Appendcols – Converging Data Sources

Examples

- **Implicit join on time**
index=* http | timechart count by sourcetype
- **Enrich data with lookup**
sourcetype=access_combined status=503
| lookup customer_info uid |
stats count by customer_value
- **Append results from another search**
... | appendcols [search earliest=-1h
sourcetype=Kepware units=W row=A
| stats stdev(Value) as hr_stdev] ...



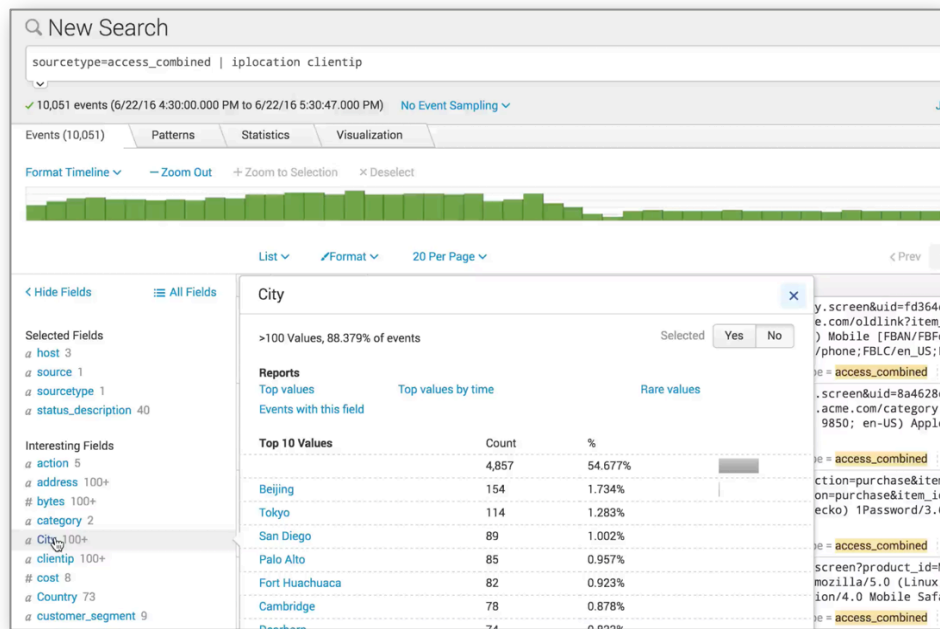
SPL Examples And Recipes

- Search and filter + creating/modifying fields
- Charting statistics and predicting values
- Enriching and converging data sources
- **Visualize geographic data in real time**
- Identifying transactions and anomalies
- Data exploration & finding relationships between fields

iplocation – Geographic Data

Examples

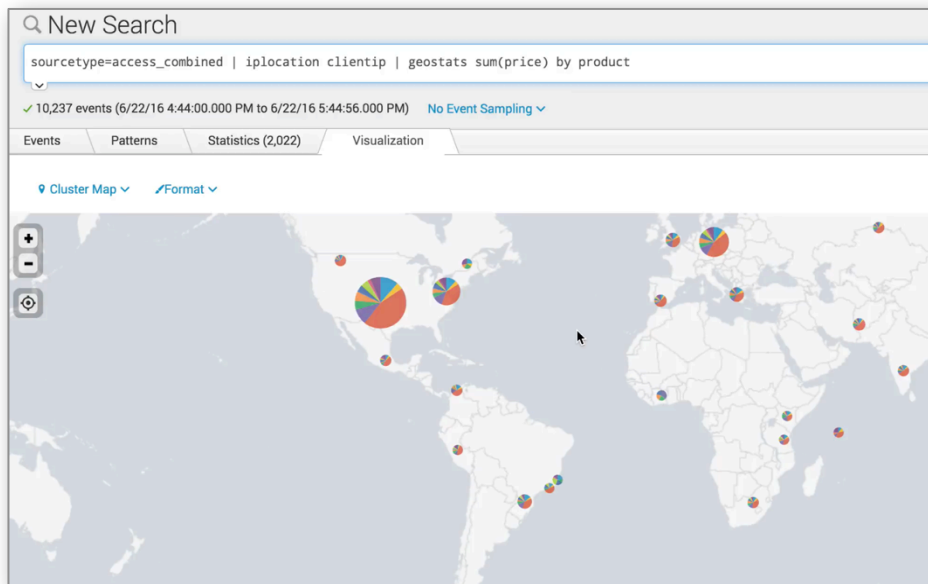
- **Assign Lat/Lon to IP addresses**
... | iplocation clientip
- Visualize statistics geographically
... | geostats sum(price) by product
- Use custom choropleths
... | geom <featureCollection> <featureId>
- Track object movements
... | table _time latitude longitude vehicleId



geostats – Geographic Data

Examples

- Assign Lat/Lon to IP addresses
... | iplocation clientip
- **Visualize statistics geographically**
... | geostats sum(price) by product
- Use custom choropleths
... | geom <featureCollection> <featureId>
- Track object movements
... | table _time latitude longitude vehicleId



geom – Geographic Data

Examples

- Assign Lat/Lon to IP addresses
... | iplocation clientip
- Visualize statistics geographically
... | geostats sum(price) by product
- **Use custom choropleths**
... | geom <featureCollection> <featureId>
- Track object movements
... | table _time latitude longitude vehicleId

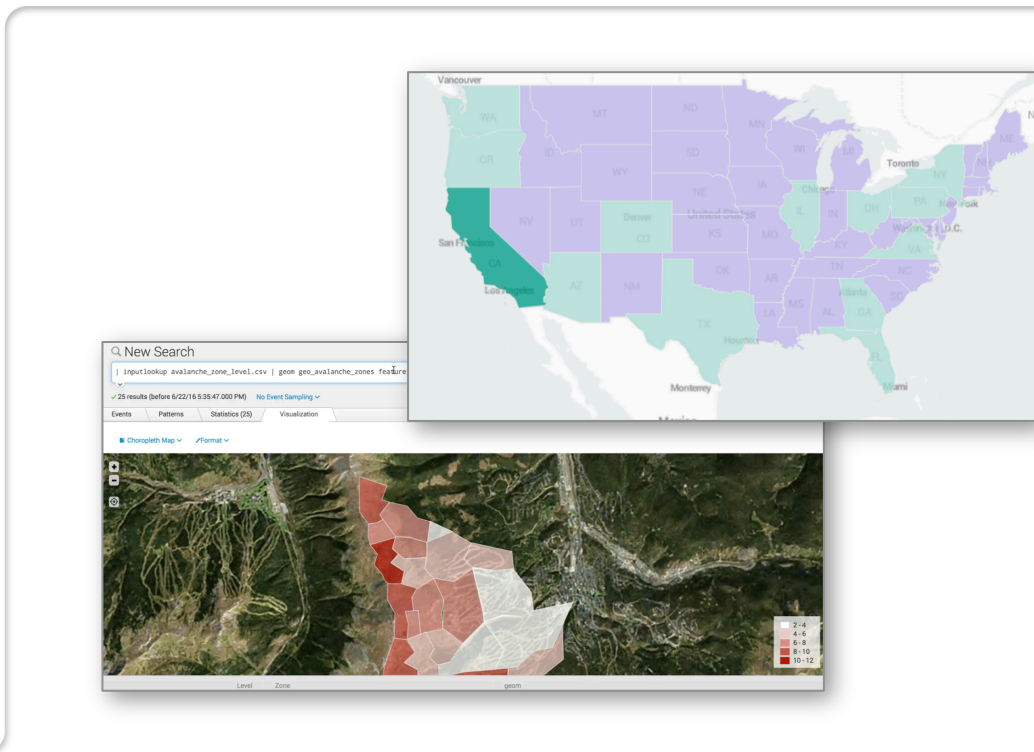
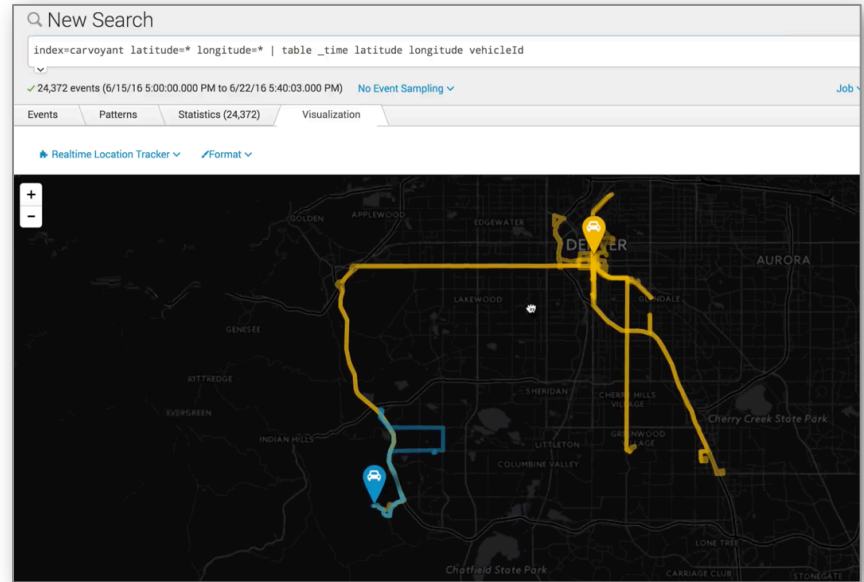


Table – Geographic Data

Examples

- Assign Lat/Lon to IP addresses
... | iplocation clientip
- Visualize statistics geographically
... | geostats sum(price) by product
- Use custom choropleths
... | geom <featureCollection> <featureId>
- **Track object movements**
... | table _time latitude longitude vehicleId



SPL Examples And Recipes

- Search and filter + creating/modifying fields
- Charting statistics and predicting values
- Enriching and Converging Data Sources
- Visualize Geographic data in real-time
- **Identifying transactions and anomalies**
- Data exploration & finding relationships between fields

Transaction – Group Related Events Spanning Time

Examples

- **Group by session ID**
`sourcetype=access*`
| `transaction JSESSIONID`
- Calculate session durations
`sourcetype=access*`
| `transaction JSESSIONID`
| `stats min(duration) max(duration)`
| `avg(duration)`
- Stats is better
`sourcetype=access*`
| `stats min(_time) AS earliest max(_time)`
| `AS latest by JSESSIONID`
| `eval duration=latest-earliest`
| `stats min(duration) max(duration)`
| `avg(duration)`

The screenshot shows a Splunk search interface with the following details:

- Search Query:** `sourcetype=access*`
| `transaction JSESSIONID`
- Results:** 247 events (7/1/14 1:15:00.000 PM to 7/1/14 2:15:08.000 PM)
- Visualization:** A bar chart showing event counts over time.
- Table View:** A table with columns for `i` (index), `Time`, and `Event`. The table shows a sequence of events for a single session ID, including actions like `purchase`, `remove`, and `view`.

i	Time	Event
>	7/1/14 1:27:34.844 PM	10.2.1.33 128.241.220.82 - - [01/Jul/2014 13:27:34:844341] "GET /cart.do?action=purchase&itemId=EST-15&product_id=MC-SANDISK-M..."
>	7/1/14 1:27:34.844 PM	10.2.1.34 62.216.64.19 - - [01/Jul/2014 13:29:41:906235] "GET /cart.do?action=changequantity&itemId=EST-16&product_id=BT-SP-JA..."
>	7/1/14 1:27:34.844 PM	10.2.1.33 12.130.60.4 - - [01/Jul/2014 13:31:59:986518] "GET /cart.do?action=purchase&itemId=EST-16&product_id=CH-APPLE-SWL&JS..."
>	7/1/14 1:27:34.844 PM	10.2.1.35 90.205.111.169 - - [01/Jul/2014 13:33:36:049273] "GET /cart.do?action=remove&itemId=EST-12&product_id=DP-HTCREZUND08..."
>	7/1/14 1:27:34.844 PM	10.2.1.34 62.216.64.19 - - [01/Jul/2014 13:39:29:250254] "GET /cart.do?action=view&itemId=EST-26&product_id=MC-SANDISK-MICROSD..."
>	7/1/14 1:27:21.837 PM	10.2.1.33 141.146.8.66 - - [01/Jul/2014 13:27:21:837389] "POST /product.screen?product_id=AC-ASSTCHARMS&JSESSIONID=SD6SL5FF2ADF3..."
>	7/1/14 1:27:21.837 PM	10.2.1.33 12.130.60.5 - - [01/Jul/2014 13:32:05:991006] "POST /product.screen?product_id=GH-APPLE-10M&JSESSIONID=SD6SL5FF2ADF3..."
>	7/1/14 1:27:21.837 PM	10.2.1.34 10.2.1.44 - - [01/Jul/2014 13:36:02:119] "POST /product.screen?product_id=CC-T10-RIM-BBERRYPLAY&JSESSIONID=SD6SL5FF2ADF3..."
>	7/1/14 1:27:21.837 PM	10.2.1.34 130.253.37.97 - - [01/Jul/2014 13:40:47:290948] "POST /product.screen?product_id=AC-SAMS-NETEXTEND&JSESSIONID=SD6SL5FF2ADF3..."
>	7/1/14 1:27:21.837 PM	10.2.1.35 131.178.233.243 - - [01/Jul/2014 13:41:19:308722] "POST /product.screen?product_id=DP-NOKLUMIA&JSESSIONID=SD6SL5FF2ADF3..."

Transaction – Group Related Events Spanning Time

Examples

- Group by session ID
`sourcetype=access*`
`| transaction JSESSIONID`
- **Calculate session durations**
`sourcetype=access*`
`| transaction JSESSIONID`
`| stats min(duration) max(duration) avg(duration)`
- Stats is better
`sourcetype=access*`
`| stats min(_time) AS earliest max(_time) AS latest by JSESSIONID`
`| eval duration=latest-earliest`
`| stats min(duration) max(duration) avg(duration)`

New Search

```
sourcetype=access*
| transaction JSESSIONID
| stats min(duration) max(duration) avg(duration)
```

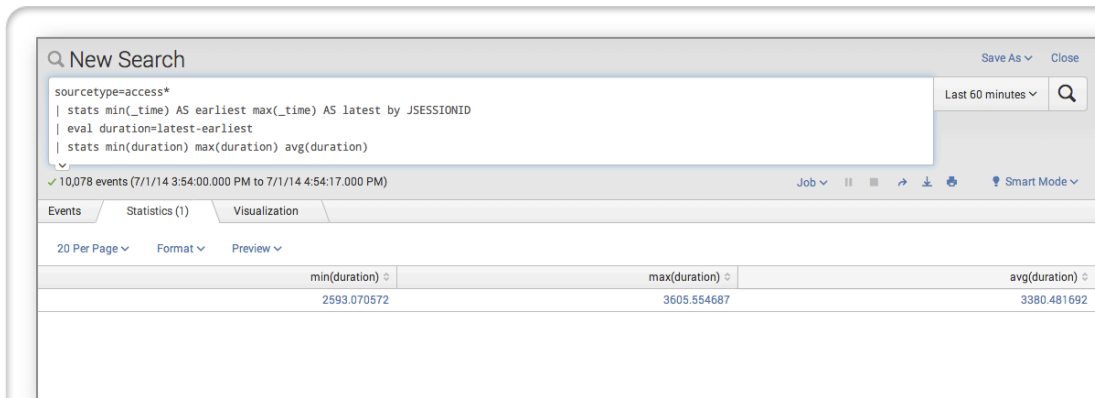
247 events (7/1/14 2:10:00.000 PM to 7/1/14 3:10:59.000 PM)

min(duration)	max(duration)	avg(duration)
2661.620435	3645.120123	3413.271055

Transaction – Group Related Events Spanning Time

Examples

- Group by session ID
sourcetype=access*
| transaction JSESSIONID
- Calculate session durations
sourcetype=access*
| transaction JSESSIONID
| stats min(duration) max(duration)
avg(duration)
- **Stats is better**
sourcetype=access*
| stats min(_time) AS earliest max(_time)
AS latest by JSESSIONID
| eval duration=latest-earliest
| stats min(duration) max(duration)
avg(duration)



The screenshot shows a Splunk search interface with the following search query:

```
sourcetype=access*
| stats min(_time) AS earliest max(_time) AS latest by JSESSIONID
| eval duration=latest-earliest
| stats min(duration) max(duration) avg(duration)
```

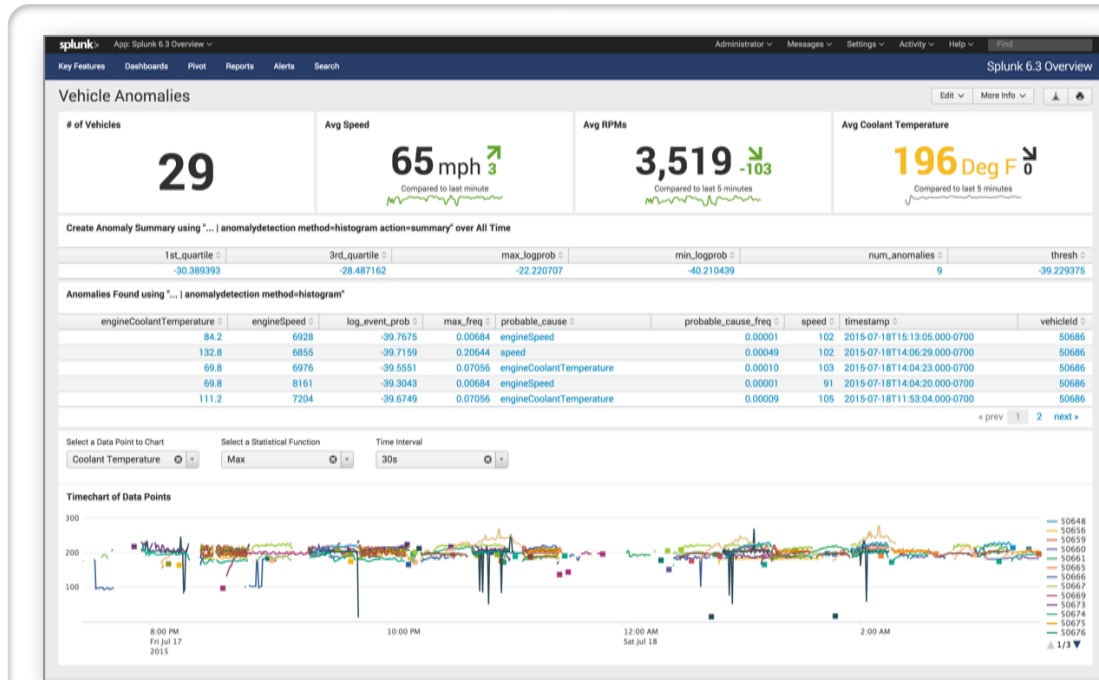
The search results show 10,078 events from 7/1/14 3:54:00.000 PM to 7/1/14 4:54:17.000 PM. The results are displayed in a table with the following columns:

min(duration)	max(duration)	avg(duration)
2593.070572	3605.554687	3380.481692

Anomaly Detection – Find Anomalies In Your Data

Examples

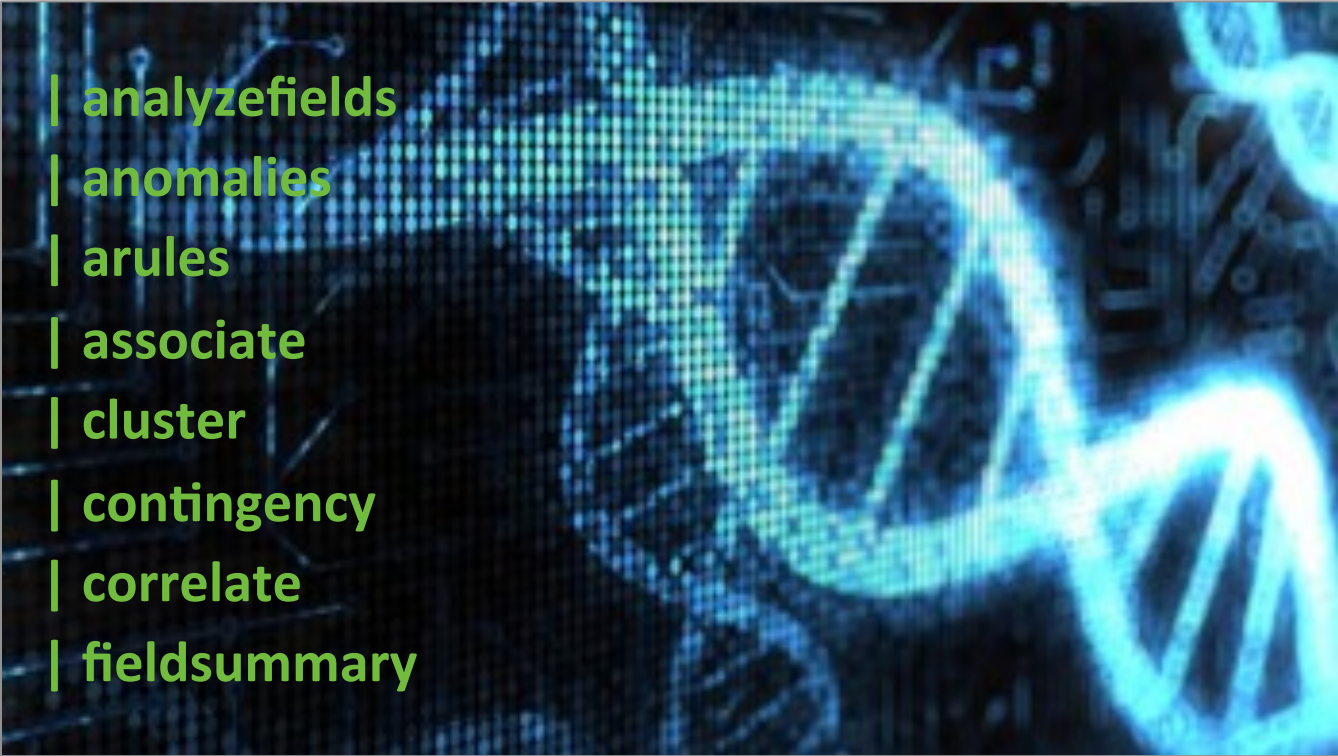
- Find anomalies
| inputlookup car_data.csv | anomalydetection
- Summarize anomalies
| inputlookup car_data.csv | anomalydetection action=summary
- Use IQR and remove outliers
| inputlookup car_data.csv | anomalydetection method=iqr action=remove



SPL Examples And Recipes

- Search and filter + creating/modifying fields
- Charting statistics and predicting values
- Enriching and Converging Data Sources
- Visualize Geographic data in real-time
- Identifying transactions and anomalies
- **Data exploration & finding relationships between fields**

Data Exploration

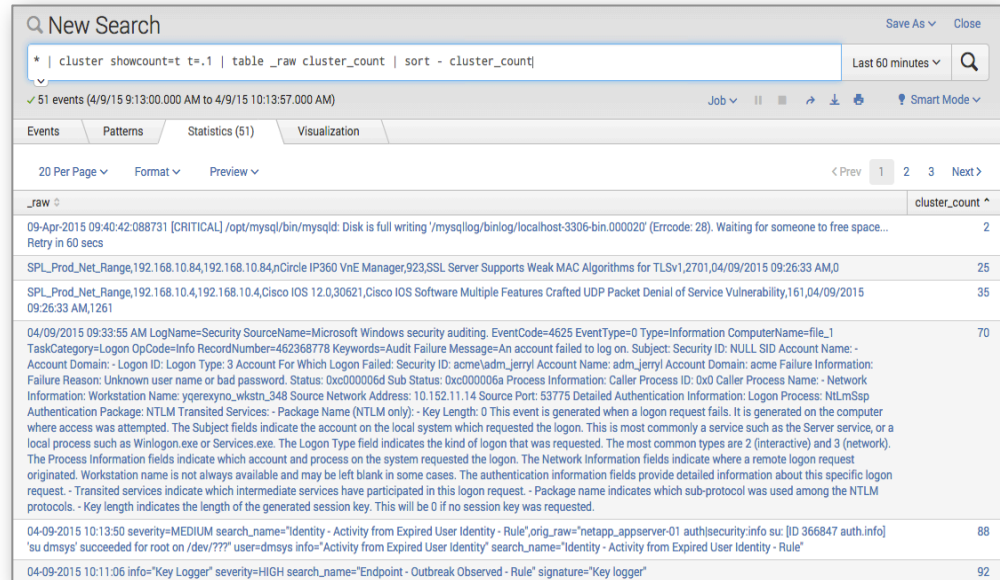


- | **analyzefields**
- | **anomalies**
- | **rules**
- | **associate**
- | **cluster**
- | **contingency**
- | **correlate**
- | **fieldsummary**

Cluster – Exploring Your Data

Examples

- **Find most/least common events**
* | cluster showcount=t t=.1
| table _raw cluster_count
- Display Summary of Fields.
sourcetype=access_combined
| fields – date* source* time*
| fieldsummary maxvals=5
- Show patterns of co-occurring fields.
sourcetype=access_combined
| fields – date* source* time* | correlate
- View field relationships
sourcetype=access_combined
| contingency uri status
- Find predictors of fields
sourcetype=access_combined
| analyzefields classfield=status



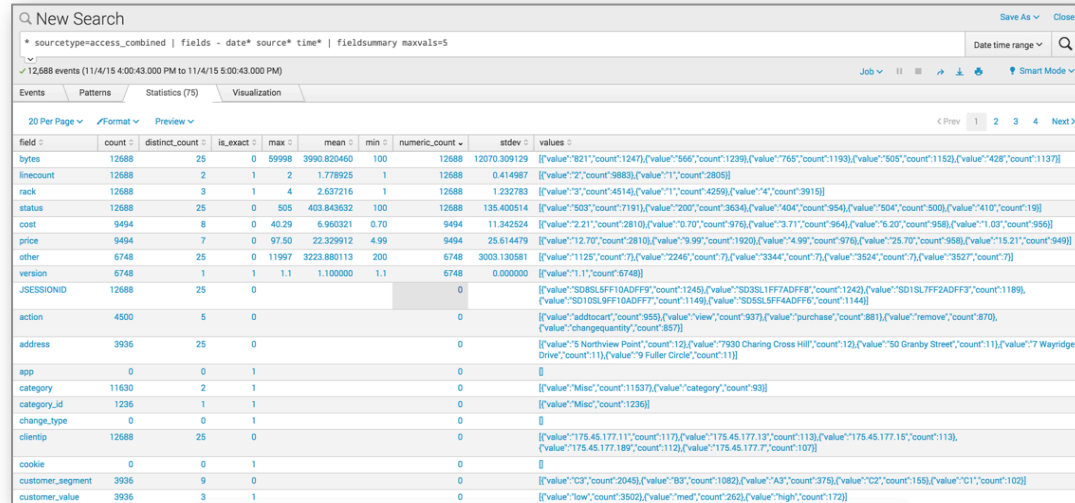
The screenshot shows a Splunk search interface with the following search query: `* | cluster showcount=t t=.1 | table _raw cluster_count | sort - cluster_count`. The results are displayed in a table with 51 events. The table has two columns: `_raw` and `cluster_count`. The events are sorted by `cluster_count` in descending order.

_raw	cluster_count
09-Apr-2015 09:40:42:088731 [CRITICAL] /opt/mysql/bin/mysqld: Disk is full writing /mysqllog/binlog/localhost-3306-bin.000020 (Errcode: 28). Waiting for someone to free space... Retry in 60 secs	2
SPL_Prod_Net_Range,192.168.10.84,192.168.10.84,nCircle IP360 VnE Manager,923,SSL Server Supports Weak MAC Algorithms for TLSv1,2701,04/09/2015 09:26:33 AM,0	25
SPL_Prod_Net_Range,192.168.10.4,192.168.10.4,Cisco IOS 12.0,30621,Cisco IOS Software Multiple Features Crafted UDP Packet Denial of Service Vulnerability,161,04/09/2015 09:26:33 AM,1261	35
04/09/2015 09:33:55 AM LogName=Security SourceName=Microsoft Windows security auditing. EventCode=4625 EventType=0 Type=Information ComputerName=file_1 TaskCategory=Logon OpCode=Info RecordNumber=462368778 Keywords=Audit Failure Message=An account failed to log on. Subject: Security ID: NULL SID Account Name: - Account Domain: - Logon ID: Logon Type: 3 Account For Which Logon Failed: Security ID: acme\adm_jerry Account Name: adm_jerry Account Domain: acme Failure Information: Failure Reason: Unknown user name or bad password. Status: 0xc000006d Sub Status: 0xc000006a Process Information: Caller Process ID: 0xd Caller Process Name: - Network Information: Workstation Name: ygeresyno_wkstn_348 Source Network Address: 10.152.11.14 Source Port: 53775 Detailed Authentication Information: Logon Process: NTLMSP Authentication Package: NTLM Transited Services: - Package Name (NTLM only): - Key Length: 0 This event is generated when a logon request fails. It is generated on the computer where access was attempted. The Subject fields indicate the account on the local system which requested the logon. This is most commonly a service such as the Server service, or a local process such as Winlogon.exe or Services.exe. The Logon Type field indicates the kind of logon that was requested. The most common types are 2 (interactive) and 3 (network). The Process Information fields indicate which account and process on the system requested the logon. The Network Information fields indicate where a remote logon request originated. Workstation name is not always available and may be left blank in some cases. The authentication information fields provide detailed information about this specific logon request. - Transited services indicate which intermediate services have participated in this logon request. - Package name indicates which sub-protocol was used among the NTLM protocols. - Key length indicates the length of the generated session key. This will be 0 if no session key was requested.	70
04-09-2015 10:13:50 severity=MEDIUM search_name="Identity - Activity from Expired User Identity - Rule",orig_raw="netapp_appserver-01 auth security:info su: [ID 366847 auth.info] 'su dmsys' succeeded for root on /dev/???' user=dmsys info="Activity from Expired User Identity" search_name="Identity - Activity from Expired User Identity - Rule"	88
04-09-2015 10:11:06 info="Key Logger" severity=HIGH search_name="Endpoint - Outbreak Observed - Rule" signature="Key logger"	92

Cluster – Exploring Your Data

Examples

- Find most/least common events
* | cluster showcount=t t=.1
| table _raw cluster_count
- Display Summary of Fields.
sourcetype=access_combined
| fields – date* source* time*
| fieldsummary maxvals=5
- Show patterns of co-occurring fields.
sourcetype=access_combined
| fields – date* source* time* | correlate
- View field relationships
sourcetype=access_combined
| contingency uri status
- Find predictors of fields
sourcetype=access_combined
| analyzefields classfield=status



The screenshot shows a Splunk search interface with the following search query: `* sourcetype=access_combined | fields - date* source* time* | fieldsummary maxvals=5`. The search results show 12,688 events from 11/4/15 4:00:43.000 PM to 11/4/15 5:00:43.000 PM. The interface includes tabs for Events, Patterns, Statistics (75), and Visualization. Below the search bar, there are controls for 20 items per page, format, and preview. The main content is a table with columns for field name, count, distinct count, is_exact, max, mean, min, numeric_count, stdev, and values. The table lists various fields such as bytes, linecount, rack, status, cost, price, other, version, JSESSIONID, action, address, app, category, category_id, change_type, clientip, cookie, customer_segment, and customer_value, each with its corresponding statistical data and a list of values.

field	count	distinct_count	is_exact	max	mean	min	numeric_count	stdev	values
bytes	12688	25	0	59998	3990.820460	100	12688	12070.309129	[[{"value": "921", "count": 1247}, {"value": "566", "count": 1239}, {"value": "766", "count": 1193}, {"value": "505", "count": 1152}, {"value": "428", "count": 1137}]]
linecount	12688	2	1	2	1.778925	1	12688	0.414987	[[{"value": "2", "count": 9883}, {"value": "1", "count": 2805}]]
rack	12688	3	1	4	2.637216	1	12688	1.232783	[[{"value": "3", "count": 4514}, {"value": "1", "count": 4259}, {"value": "4", "count": 3915}]]
status	12688	25	0	505	403.843632	100	12688	135.400514	[[{"value": "503", "count": 7191}, {"value": "200", "count": 3634}, {"value": "404", "count": 954}, {"value": "504", "count": 800}, {"value": "410", "count": 19}]]
cost	9494	8	0	40.29	6.960321	0.70	9494	11.342524	[[{"value": "2.21", "count": 2810}, {"value": "0.70", "count": 976}, {"value": "3.71", "count": 954}, {"value": "6.20", "count": 958}, {"value": "1.03", "count": 956}]]
price	9494	7	0	97.50	22.329912	4.99	9494	25.614479	[[{"value": "12.70", "count": 2810}, {"value": "9.99", "count": 1920}, {"value": "4.99", "count": 976}, {"value": "26.70", "count": 958}, {"value": "15.21", "count": 949}]]
other	6748	25	0	11997	3223.880113	200	6748	3003.130581	[[{"value": "1125", "count": 7}, {"value": "2246", "count": 7}, {"value": "3344", "count": 7}, {"value": "3524", "count": 7}, {"value": "3527", "count": 7}]]
version	6748	1	1	1.1	1.100000	1.1	6748	0.000000	[[{"value": "1.1", "count": 6748}]]
JSESSIONID	12688	25	0				0		[[{"value": "SD8SL5FF10ADF9", "count": 1245}, {"value": "SD3SL1FF7ADF8", "count": 1242}, {"value": "SD1SL7FF2ADF3", "count": 1189}, {"value": "SD10SL9FF10ADF7", "count": 1149}, {"value": "SD6SL5FF4ADF6", "count": 1144}]]
action	4500	5	0				0		[[{"value": "addtocart", "count": 959}, {"value": "view", "count": 937}, {"value": "purchase", "count": 881}, {"value": "remove", "count": 870}, {"value": "changequantity", "count": 357}]]
address	3936	25	0				0		[[{"value": "3 Northview Point", "count": 12}, {"value": "7930 Charing Cross Hill", "count": 12}, {"value": "50 Granby Street", "count": 11}, {"value": "7 Wayridge Drive", "count": 11}, {"value": "9 Fuller Circle", "count": 11}]]
app	0	0	1				0		[[{"value": "", "count": 0}]]
category	11630	2	1				0		[[{"value": "Misc", "count": 1153}, {"value": "category", "count": 93}]]
category_id	1236	1	1				0		[[{"value": "Misc", "count": 1236}]]
change_type	0	0	1				0		[[{"value": "", "count": 0}]]
clientip	12688	25	0				0		[[{"value": "175.45.177.11", "count": 117}, {"value": "175.45.177.13", "count": 113}, {"value": "175.45.177.15", "count": 113}, {"value": "175.45.177.189", "count": 112}, {"value": "175.45.177.7", "count": 107}]]
cookie	0	0	1				0		[[{"value": "", "count": 0}]]
customer_segment	3936	9	0				0		[[{"value": "CS", "count": 2045}, {"value": "B3", "count": 1082}, {"value": "A3", "count": 375}, {"value": "C2", "count": 155}, {"value": "C1", "count": 102}]]
customer_value	3936	3	1				0		[[{"value": "low", "count": 3502}, {"value": "med", "count": 262}, {"value": "high", "count": 172}]]

Correlate – Exploring Your Data

Examples

- Find most/least common events
* | cluster showcount=t t=.1
| table _raw cluster_count
- Display Summary of Fields.
sourcetype=access_combined
| fields – date* source* time*
| fieldsummary maxvals=5
- **Show patterns of co-occurring fields.**
sourcetype=access_combined
| fields – date* source* time* | correlate
- View field relationships
sourcetype=access_combined
| contingency uri status
- Find predictors of fields
sourcetype=access_combined
| analyzefields classfield=status

The screenshot shows a Splunk search interface with the following search query: `sourcetype=access_combined | fields - date* source* time* | correlate`. The search results show 10,490 events. Below the search bar, there are tabs for 'Events', 'Patterns', 'Statistics (60)', and 'Visualization'. The 'Statistics' tab is active, displaying a correlation matrix table with 11 columns representing different fields: JSESSIONID, action, address, bytes, category, category_id, clientip, cost, customer_segment, customer_value, and customer_wallet. Each cell in the matrix contains a numerical value representing the correlation coefficient between the fields.

RowField	JSESSIONID	action	address	bytes	category	category_id	clientip	cost	customer_segment	customer_value	customer_wallet
JSESSIONID	1.00	0.34	0.28	1.00	0.76	0.10	1.00	0.76	0.28	0.28	0.28
action	0.34	1.00	0.17	0.34	0.45	0.00	0.34	0.45	0.17	0.17	0.17
address	0.28	0.17	1.00	0.28	0.16	0.00	0.28	0.16	1.00	1.00	1.00
bytes	1.00	0.34	0.28	1.00	0.76	0.10	1.00	0.76	0.28	0.28	0.28
category	0.76	0.45	0.16	0.76	1.00	0.00	0.76	1.00	0.16	0.16	0.16
category_id	0.10	0.00	0.00	0.10	0.00	1.00	0.10	0.00	0.00	0.00	0.00
clientip	1.00	0.34	0.28	1.00	0.76	0.10	1.00	0.76	0.28	0.28	0.28
cost	0.76	0.45	0.16	0.76	1.00	0.00	0.76	1.00	0.16	0.16	0.16
customer_segment	0.28	0.17	1.00	0.28	0.16	0.00	0.28	0.16	1.00	1.00	1.00
customer_value	0.28	0.17	1.00	0.28	0.16	0.00	0.28	0.16	1.00	1.00	1.00
customer_wallet	0.28	0.17	1.00	0.28	0.16	0.00	0.28	0.16	1.00	1.00	1.00
description	0.76	0.45	0.16	0.76	1.00	0.00	0.76	1.00	0.16	0.16	0.16
device	0.59	0.54	0.30	0.59	0.63	0.00	0.59	0.63	0.30	0.30	0.30
email	0.28	0.17	1.00	0.28	0.16	0.00	0.28	0.16	1.00	1.00	1.00

Contingency – Exploring Your Data

Examples

- Find most/least common events
* | cluster showcount=t t=.1
| table _raw cluster_count
- Display Summary of Fields.
sourcetype=access_combined
| fields – date* source* time*
| fieldsummary maxvals=5
- Show patterns of co-occurring fields.
sourcetype=access_combined
| fields – date* source* time* | correlate
- **View field relationships**
sourcetype=access_combined
| contingency uri status
- Find predictors of fields
sourcetype=access_combined
| analyzefields classfield=status

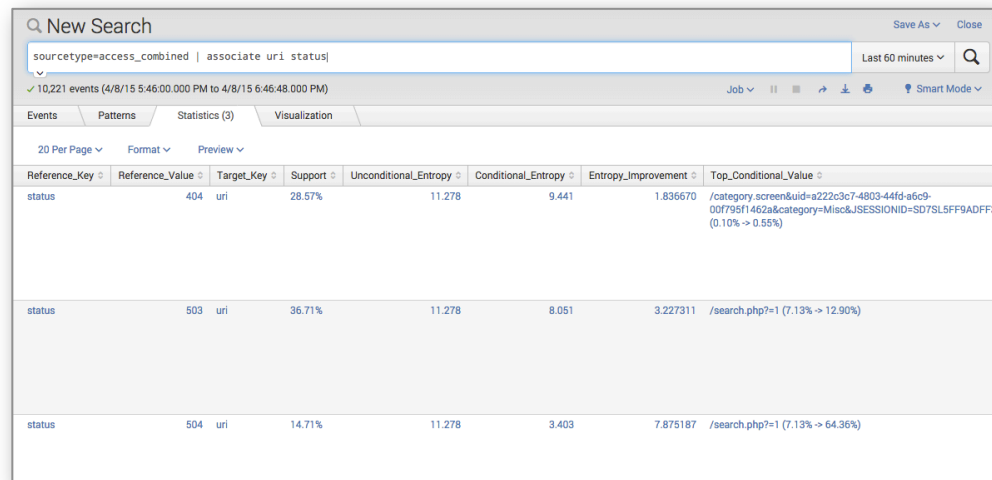
The screenshot shows a Splunk search interface with the following search query: `sourcetype=access_combined | contingency uri status`. The results are displayed as a table with 10 columns representing different status values (503, 200, 404, 504, 415, 100, 303, 405, 401, 201, 411, 409, 403, 206) and 8 rows representing different URI values. The table shows the count of events for each combination of URI and status.

uri	503	200	404	504	415	100	303	405	401	201	411	409	403	206
/search.php?=-1	119	0	0	238	0	0	0	0	0	0	0	0	0	0
/search.php?!(SELECT	119	0	0	0	0	0	0	0	0	0	0	0	0	0
/category.screen&uid=13ecd676-1aac-489c-9273-64f05beaded3&category=Misc&JSESSIONID=SD7SL7FF6ADFF5	0	4	3	0	0	0	0	0	0	0	0	0	0	0
/category.screen&uid=a8310182-8593-4092-8687-5508a01f3901&category=Misc&JSESSIONID=SD2SBL2FF1ADFF4	0	5	1	0	0	0	0	0	0	0	0	0	0	0
/category.screen&uid=efd2eb40-2ef3-418f-9173-6ce44112c842&category=Misc&JSESSIONID=SD1SBL2FF7ADFF8	0	5	0	0	0	0	0	0	0	0	0	0	0	0
/category.screen&uid=c2ee77ba-f435-470e-9d5e-bfddf5862000&category=Misc&JSESSIONID=SD9SL7FF2ADFF9	0	2	3	0	0	0	0	0	0	0	0	0	0	0
/category.screen&uid=a222c3c7-4803-44fd-a6c9-00f795f1462a&category=Misc&JSESSIONID=SD7SL5FF9ADFF3	0	1	4	0	0	0	0	0	0	0	0	0	0	0
/category.screen&uid=a1395aac-e835-4f42-89ae-a659b89919c7&category=Misc&JSESSIONID=SD1SBL1FF6ADFF2	0	5	0	0	0	0	0	0	0	0	0	0	0	0
/category.screen&uid=7a47ef03-64ac-40db-8f9b-c549698d30b9&category=Misc&JSESSIONID=SD5SBL1FF1ADFF8	0	4	0	0	0	0	0	0	0	0	0	0	0	0
/category.screen&uid=10c1fd9c-d2a3-48da-9fb1-29f59d52bd3d&category=Misc&JSESSIONID=SD1SCL1FF10ADFF6	0	5	0	0	0	0	0	0	0	0	0	0	0	0

Analyzefields – Exploring Your Data

Examples

- Find most/least common events
* | cluster showcount=t t=.1
| table _raw cluster_count
- Display Summary of Fields.
sourcetype=access_combined
| fields – date* source* time*
| fieldsummary maxvals=5
- Show patterns of co-occurring fields.
sourcetype=access_combined
| fields – date* source* time* | correlate
- View field relationships
sourcetype=access_combined
| contingency uri status
- **Find predictors of fields**
sourcetype=access_combined
| analyzefields classfield=status



New Search

sourcetype=access_combined | associate uri status

10,221 events (4/8/15 5:46:00.000 PM to 4/8/15 6:46:48.000 PM)

Events Patterns Statistics (3) Visualization

20 Per Page Format Preview

Reference_Key	Reference_Value	Target_Key	Support	Unconditional_Entropy	Conditional_Entropy	Entropy_Improvement	Top_Conditional_Value
status	404	uri	28.57%	11.278	9.441	1.836670	/category.screen&uid=a222c3c7-4803-44fd-a6c9-00f795f1462a&category=Misc&JSESSIONID=SD7SL5FF9ADFF3 (0.10% -> 0.55%)
status	503	uri	36.71%	11.278	8.051	3.227311	/search.php?r=1 (7.13% -> 12.90%)
status	504	uri	14.71%	11.278	3.403	7.875187	/search.php?r=1 (7.13% -> 64.36%)

Machine Learning Toolkit And Showcase

Examples

- Predict Numeric Fields
- Predict Categorical Fields
- Detect Numerical Outliers
- Detect Categorical Outliers
- Forecast Time Series
- Cluster Events

Welcome to the Machine Learning Toolkit and Showcase. Click on the dashboards or examples below to explore the kinds of analytics this app enables. Each dashboard includes both end-to-end examples with datasets we have provided, as well as the ability to apply the dashboard to your own data. You can inspect the dashboard panels and other code to see how each one works and then create custom dashboards to suit your needs. Everything you see was implemented on the Splunk platform using public interfaces, so you can bring similar functionality to your own organization's Splunk instance; there's nothing hidden up our sleeves.

Predict Numeric Fields

Predict the value of a numeric field using a weighted combination of the values of other fields in that event. A common use of these predictions is to identify anomalies: predictions that differ significantly from the actual value may be considered anomalous. IT admins could predict the values of sensors that were malfunctioning; security analysts could predict how much data a user is likely to transfer and flag unusually high prediction errors; and business analysts could predict the likely spending habits of customers.

Algorithm: Linear Regression

Examples:

- Predict Median House Value
- Predict Baseball Runs
- Predict App Usage from Other Apps

Predict Categorical Fields

Predict the value of a categorical field using the values of other fields in that event. A common use of these predictions is to identify anomalies: high-confidence predictions that turn out to be incorrect may be considered anomalous. IT admins could predict the correct values of missing configuration variables; security analysts could predict what actions a user is likely to perform and raise an alert when the user behaves unexpectedly; and business analysts could predict customer churn based on other factors.

Algorithm: Logistic Regression

Examples:

- Predict Telecom Customer Churn
- Predict Species of Iris from Physical Measurements
- Predict Incidence of Diabetes from Health Metrics

Detect Numeric Outliers

Find values that are far from previous values. IT admins could look for machines with unusually high resource utilization; security analysts could look for employees transferring unusually large amounts of data; and business analysts could identify big spenders.

Algorithm: Distribution statistics

Examples:

- Detect Outliers in Server Response Time

Detect Categorical Outliers

Find events that contain unusual combinations of values. IT admins could look for unusual machine configurations; security analysts could look for employees performing an atypical combination of activities; and business analysts could identify rare purchasing habits.

Algorithm: Probabilistic measures

Examples:

- Detect Outliers in Mortgage Contract Data
- Detect Outliers in Congressional Voting Records

174.51.216.216:8090/en-US/app/Splunk_ML_Toolkit/showcase_classification?ml_toolkit_dataset=Diab

Custom Commands

.conf2016

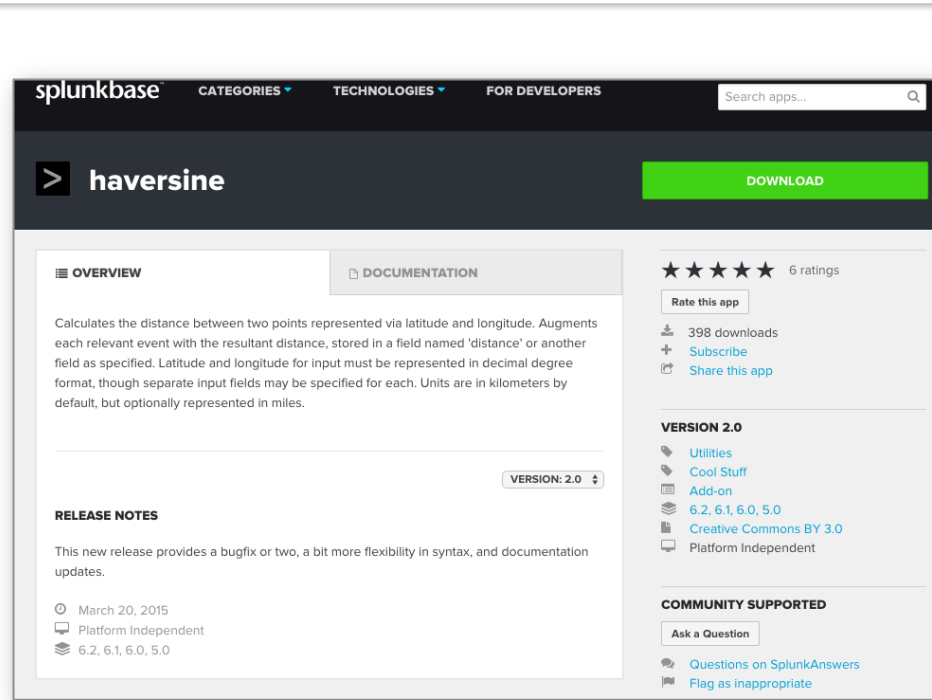
Custom Commands

- What is a Custom Command?
 - “| **haversine** origin="47.62,-122.34" outputField=dist lat lon”
- Why do we use Custom Commands?
 - Run other/external algorithms on your Splunk data
 - Save time munging data (see Timewrap!)
 - Because you can!
- Create your own or download as Apps
 - [Haversine](#) (Distance between two GPS coords)
 - [Timewrap](#) (Enhanced Time overlay)
 - [Levenshtein](#) (Fuzzy string compare)
 - [Base64](#) (Encode/Decode)

Custom Commands – Haversine

Examples

- **Download and install App [Haversine](#)**
- *Read documentation then use in SPL!*
`sourcetype=access*`
`| iplocation clientip`
`| search City=A*`
`| haversine origin="47.62,-122.34"`
`units=mi outputField=dist lat lon`
`| table clientip, City, dist, lat, lon`

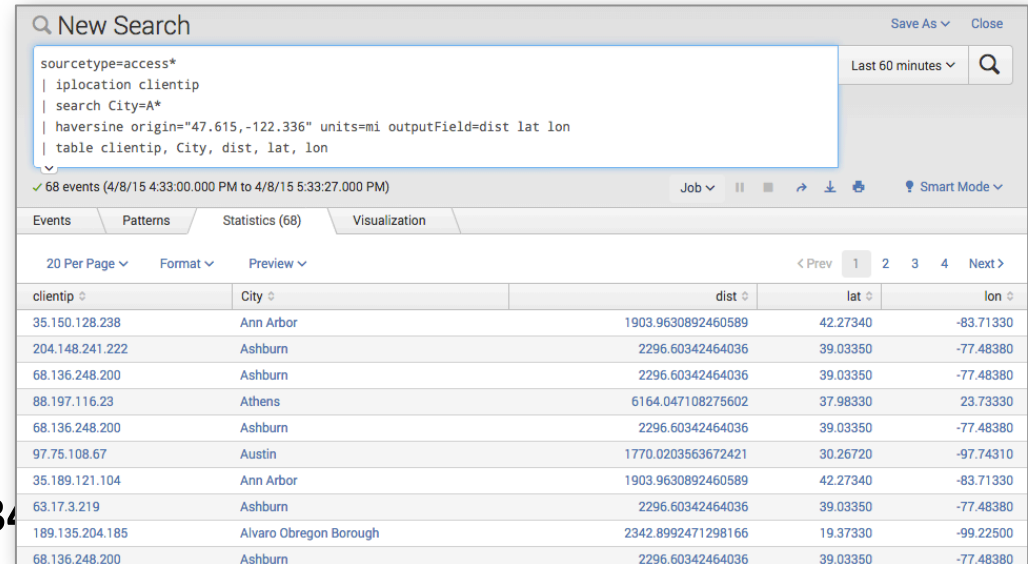


Custom Commands – Haversine

Examples

- Download and install App [Haversine](#)
- ***Read documentation then use in SPL!***

```
sourcetype=access*
| iplocation clientip
| search City=A*
| haversine origin="47.62,-122.34"
units=mi outputField=dist lat lon
| table clientip, City, dist, lat, lon
```



New Search

```
sourcetype=access*
| iplocation clientip
| search City=A*
| haversine origin="47.615,-122.336" units=mi outputField=dist lat lon
| table clientip, City, dist, lat, lon
```

68 events (4/8/15 4:33:00.000 PM to 4/8/15 5:33:27.000 PM)

clientip	City	dist	lat	lon
35.150.128.238	Ann Arbor	1903.9630892460589	42.27340	-83.71330
204.148.241.222	Ashburn	2296.60342464036	39.03350	-77.48380
68.136.248.200	Ashburn	2296.60342464036	39.03350	-77.48380
88.197.116.23	Athens	6164.047108275602	37.98330	23.73330
68.136.248.200	Ashburn	2296.60342464036	39.03350	-77.48380
97.75.108.67	Austin	1770.0203563672421	30.26720	-97.74310
35.189.121.104	Ann Arbor	1903.9630892460589	42.27340	-83.71330
63.17.3.219	Ashburn	2296.60342464036	39.03350	-77.48380
189.135.204.185	Alvaro Obregon Borough	2342.8992471298166	19.37330	-99.22500
68.136.248.200	Ashburn	2296.60342464036	39.03350	-77.48380

For More Information

- Additional information can be found in:
 - [Search Manual](#)
 - [Blogs](#)
 - [Answers](#)
 - Operational Intelligence Cookbook – [available for purchase](#)
 - [Exploring Splunk](#)

What Now?

Related breakout sessions and activities...

- Power of SPL Booth

Q & A

.conf2016

splunk >

THANK YOU

.conf2016