Dashboard Time Selection

Balancing flexibility with a series of system-crushing searches

Chuck Gilbert | Analyst, chuck_gilbert@comcast.com

September 2017 | Washington, DC
Forward-Looking Statements

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

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

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

▶ How users select time

▶ Problem statement

▶ Proposed solutions (with sample XML)
The Splunk Time Picker Input
A thing of beauty

- Flexible
- Easy to use
- Easy to set up
Problem Statement

It can result in expensive searches

Why is this a problem?

- It allows selecting data across “all time”
- You may have hundreds of users
- You may have hundreds of dashboards

What could happen?

- Expensive searches can overload the system
- Dashboards can take an inconveniently long time to populate (e.g., minutes or hours to complete)
Proposed Solution

Predetermined time intervals

- Provide a set of predetermined time intervals that serve all users’ needs

- For longer running searches, use a saved, scheduled search to precompute and cache the results. This gives the users a responsive, fast loading dashboard
Three Implementation Options

1. Use multiple panels that are alternately hidden or displayed, or

2. Cache a bigger, more detailed result set, then call only a subset of the data, or

3. Use the standard Splunk time picker, but check the duration selected by the user and respond appropriately
In all three implementation options, we detect which time period was selected by the user.

After detecting the user selection, we can set/unset tokens to customize searches or show/hide objects.
Commonality: Sample XML Period

Shared XML common to all three proposals

```xml
<change>
  <condition value="RT"> ... </condition>

  <condition value="24h">
    <set token="globalTime_tok.earliest">-24h@h</set>
    <set token="span_tok">15m</set>
    <unset token="RT_tok"></unset>
  </condition>

  <condition value="7d"> ... </condition>

  <condition value="28d">
    <set token="globalTime_tok.earliest">-28d@d</set>
    <set token="span_tok">1h</set>
    <unset token="RT_tok"></unset>
  </condition>
</change>
```

Splunk Documentation

http://docs.splunk.com/Documentation/Splunk/6.6.3/Viz/tokens#Conditional_operations_with_form_inputs
Option #1: Hidden Objects

The brute force approach

Dashed outline indicates a “hidden” object
Option #1: Hidden Objects

Sample XML to show/hide panels

```xml
<row>
  <panel depends="$RT_tok$">
    <title>This panel uses a run-time query</title>
  </panel>
  <panel rejects="$RT_tok$">
    <title>This panel displays a cached result</title>
  </panel>
</row>
```

Splunk Documentation
http://docs.splunk.com/Documentation/Splunk/6.6.3/Viz/tokens#Access_tokens_to_show_or_hide_user_interface_components
Option #2: Cache a Bigger Result Set
Finding balance between having one cached result set versus multiple

- Useful when all of the use cases can be pre-computed (and there is no need for near real time data)

- The idea is to simplify by having fewer scheduled, saved searches, then write your search to pull only a subset of the cached results
Option #2: Cache a Bigger Result Set

Sample SPL

```plaintext
| loadjob savedsearch= "myusername:search:My Saved Search"
| loadjob savedsearch= "myusername:search:My Saved Search"
| where _time < relative_time(now(),"-6d@d")
AND _time > relative_time(now(),"-7d@d")
```

Splunk Documentation
http://docs.splunk.com/Documentation/Splunk/6.6.3/SearchReference/DateandTimeFunctions#relative_time.28X.2CY.29
Option #3: Standard Time Picker Input

Use the standard time input, but test selected period

- **Standard Splunk time input**

  - Less than 24h
  - More than 7d
  - 1d < t < 7d

  - Run SPL (-24h) [then auto-refresh]
  - Fetch a 7 day cached result set
  - Fetch a 28-day cached result set
Option #3: Standard Time Picker Input

Sample XML

```xml
<change>
  <condition match="(relative_time(now(), $time_tok.latest$) -
               relative_time(now(), $time_tok.earliest$))
               &lt;= 86400">
    <!-- If selected time spans < 1d, run real-time query. -->
    <set token="less_than_1_day">true</set>
    <set token="short-ish">true</set>
    <unset token="long-ish"></unset>
  </condition>

  <condition match="(relative_time(now(), $time_tok.latest$) -
                   relative_time(now(), $time_tok.earliest$))
                   &gt; 86400">
    <!-- If selected time spans > 1d, pull from cached data. -->
    <set token="more_than_1_day">true</set>
    <set token="long-ish">true</set>
    <unset token="short-ish"></unset>
  </condition>
</change>
```

Splunk Documentation
http://docs.splunk.com/Documentation/Splunk/6.6.3/Viz/tokens
#Search_tokens_for_dynamic_display_example
Many Other Options as Well

- Post-process searches
- Report acceleration
- Dedicated summary indexes
- Data models
- Pivot tables
1. Splunk has a wide variety of tools to speed up expensive searches

2. Even if you don’t have the permissions or expertise to do the first thing you think of, you probably still have several other options
Thank You

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