Dashboard Time Selection

Balancing flexibility with a series of system-crushing searches

Chuck Gilbert | Analyst, chuck_gilbert@comcast.com

September 2017 | Washington, DC

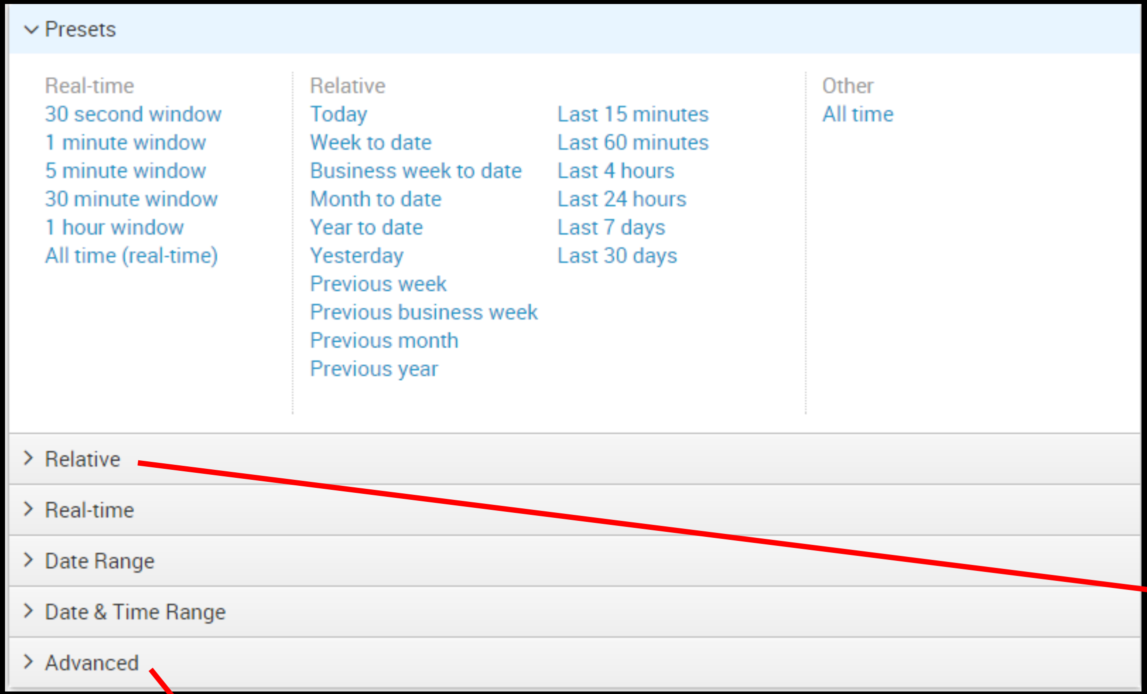
Forward-Looking Statements

Agenda

- How users select time
- Problem statement
- Proposed solutions (with sample XML)

The Splunk Time Picker Input

A thing of beauty



Presets

Real-time	Relative		Other
30 second window	Today	Last 15 minutes	All time
1 minute window	Week to date	Last 60 minutes	
5 minute window	Business week to date	Last 4 hours	
30 minute window	Month to date	Last 24 hours	
1 hour window	Year to date	Last 7 days	
All time (real-time)	Yesterday	Last 30 days	
	Previous week		
	Previous business week		
	Previous month		
	Previous year		

Relative

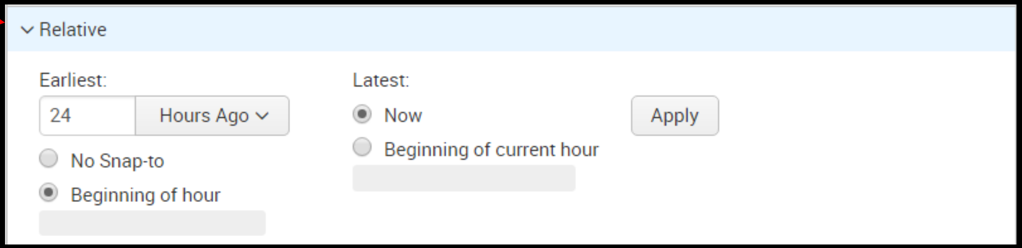
Real-time

Date Range

Date & Time Range

Advanced

- Flexible
- Easy to use
- Easy to set up



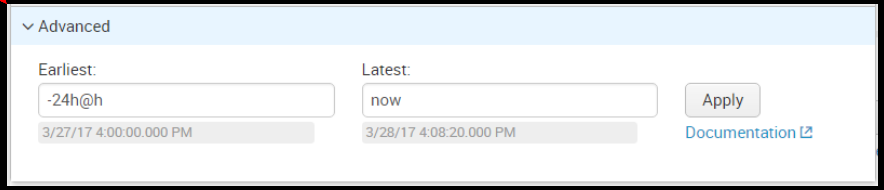
Relative

Earliest: 24 Hours Ago

No Snap-to [✓ Beginning of hour]

Latest: [✓ Now] Beginning of current hour

Apply



Advanced

Earliest: -24h@h
3/27/17 4:00:00.000 PM

Latest: now
3/28/17 4:08:20.000 PM

Apply

Documentation

splunk> .conf2017

Problem Statement

It can result in expensive searches

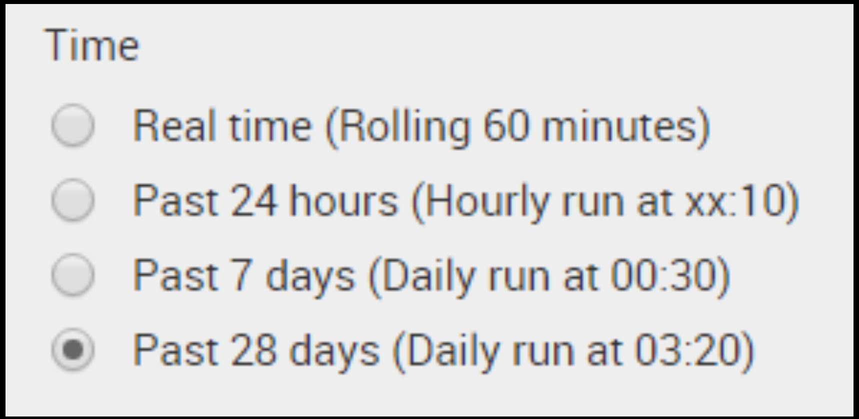
- Why is this a problem?
  - It allows selecting data across “all time”
  - You may have hundreds of users
  - You may have hundreds of dashboards
- What could happen?
  - Expensive searches can overload the system
  - Dashboards can take an inconveniently long time to populate (e.g., minutes or hours 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

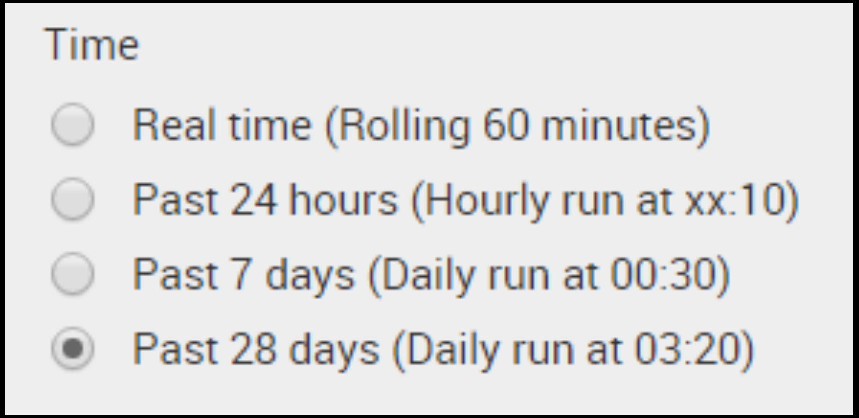


Time

- ○ Real time (Rolling 60 minutes)
- ○ Past 24 hours (Hourly run at xx:10)
- ○ Past 7 days (Daily run at 00:30)
- [✓ ● Past 28 days (Daily run at 03:20)]

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

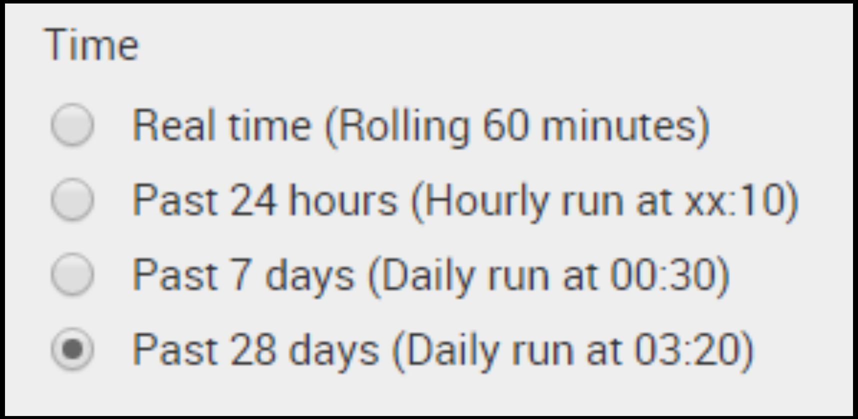

Time

- ○ Real time (Rolling 60 minutes)
- ○ Past 24 hours (Hourly run at xx:10)
- ○ Past 7 days (Daily run at 00:30)
- ● Past 28 days (Daily run at 03:20)

Commonality

Points common to all three proposals

- 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


Time

- ○ Real time (Rolling 60 minutes)
- ○ Past 24 hours (Hourly run at xx:10)
- ○ Past 7 days (Daily run at 00:30)
- ● Past 28 days (Daily run at 03:20)

Commonality: Sample XML Period

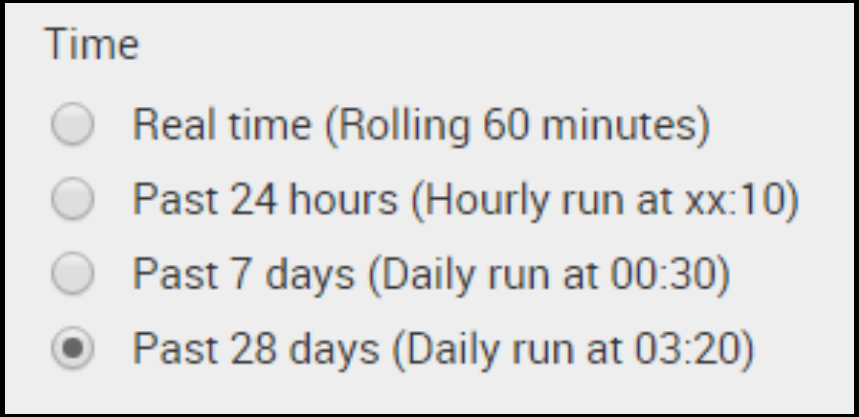
Shared XML common to all three proposals

```
<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>
</input>
```


Time

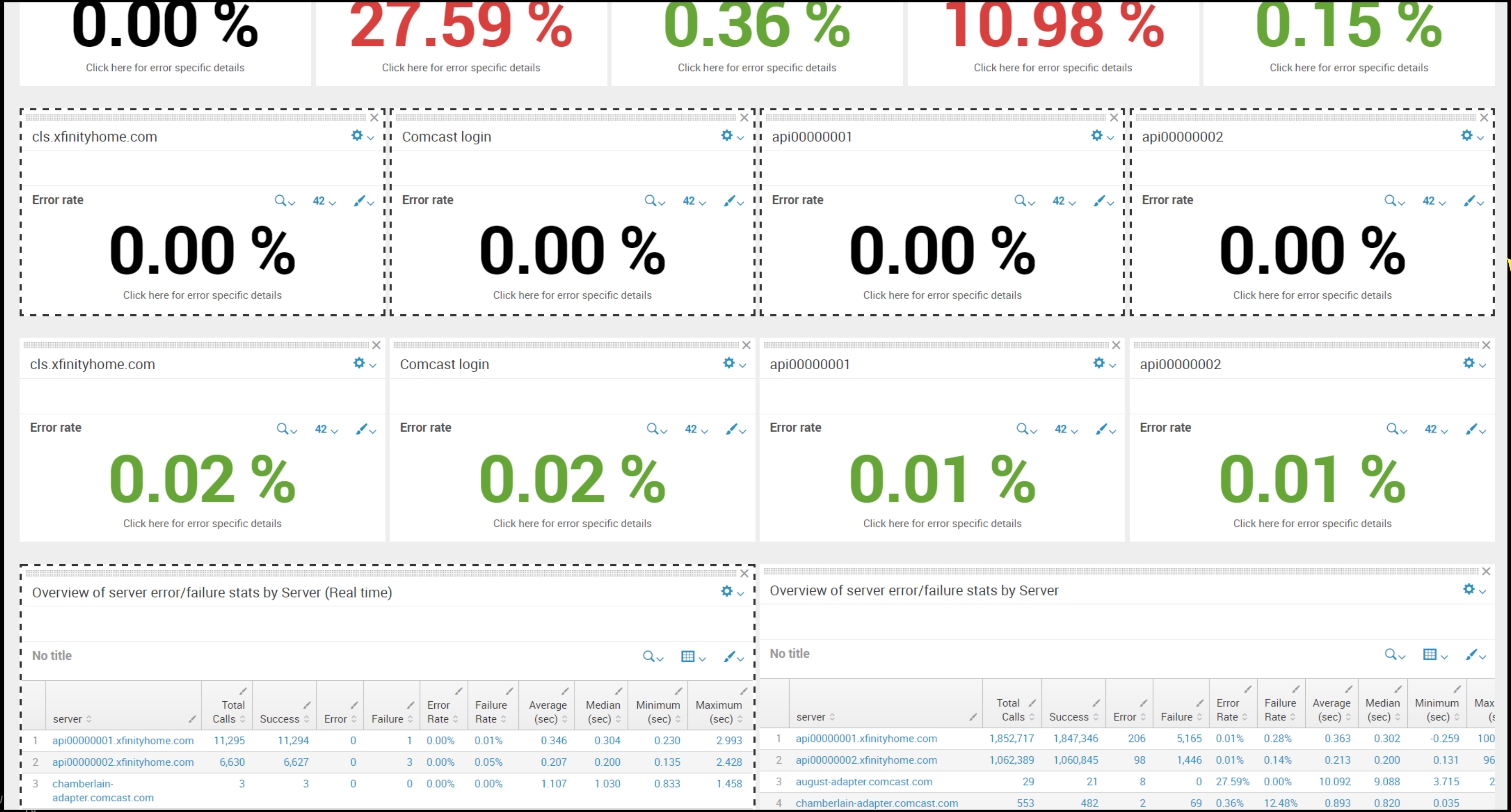
- [ ] Real time (Rolling 60 minutes)
- [ ] Past 24 hours (Hourly run at xx:10)
- [ ] Past 7 days (Daily run at 00:30)
- [✓ Past 28 days (Daily run at 03:20)]

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



0.00 %
Click here for error specific details

27.59 %
Click here for error specific details

0.36 %
Click here for error specific details

10.98 %
Click here for error specific details

0.15 %
Click here for error specific details

cls.xfinityhome.com
Error rate
0.00 %
Click here for error specific details

Comcast login
Error rate
0.00 %
Click here for error specific details

api00000001
Error rate
0.00 %
Click here for error specific details

api00000002
Error rate
0.00 %
Click here for error specific details

cls.xfinityhome.com
Error rate
0.02 %
Click here for error specific details

Comcast login
Error rate
0.02 %
Click here for error specific details

api00000001
Error rate
0.01 %
Click here for error specific details

api00000002
Error rate
0.01 %
Click here for error specific details

Overview of server error/failure stats by Server (Real time)

No title

	server	Total Calls	Success	Error	Failure	Error Rate	Failure Rate	Average (sec)	Median (sec)	Minimum (sec)	Maximum (sec)
1	api00000001.xfinityhome.com	11,295	11,294	0	1	0.00%	0.01%	0.346	0.304	0.230	2.993
2	api00000002.xfinityhome.com	6,630	6,627	0	3	0.00%	0.05%	0.207	0.200	0.135	2.428
3	chamberlain-adapter.comcast.com	3	3	0	0	0.00%	0.00%	1.107	1.030	0.833	1.458

Overview of server error/failure stats by Server

No title

	server	Total Calls	Success	Error	Failure	Error Rate	Failure Rate	Average (sec)	Median (sec)	Minimum (sec)	Max
1	api00000001.xfinityhome.com	1,852,717	1,847,346	206	5,165	0.01%	0.28%	0.363	0.302	-0.259	100
2	api00000002.xfinityhome.com	1,062,389	1,060,845	98	1,446	0.01%	0.14%	0.213	0.200	0.131	96
3	august-adapter.comcast.com	29	21	8	0	27.59%	0.00%	10.092	9.088	3.715	2
4	chamberlain-adapter.comcast.com	553	482	2	69	0.36%	12.48%	0.893	0.820	0.035	

Dashed outline indicates a “hidden” object

Option #1: Hidden Objects

Sample XML to show/hide panels

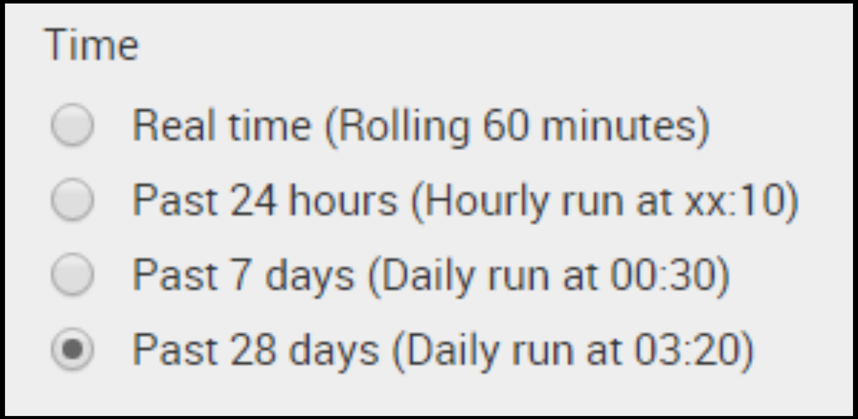
```
<row>
  <panel depends="$RT_tok$">
    <title>This panel uses a run-time
query</title>
  </panel>
  <panel rejects="$RT_tok$">
    <title>This panel displays a cached
result</title>
  </panel>
</row>
```

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

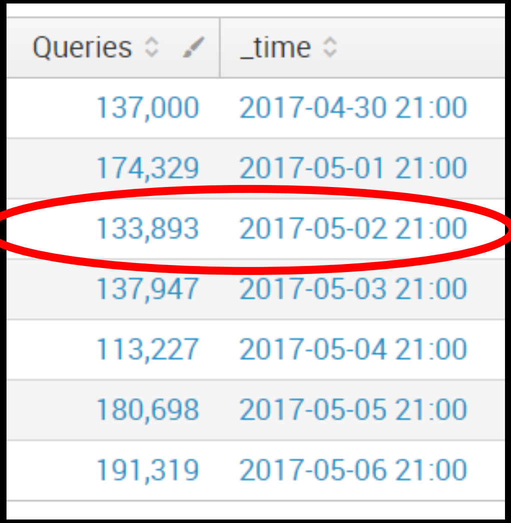

Time

- ○ Real time (Rolling 60 minutes)
- ○ Past 24 hours (Hourly run at xx:10)
- ○ Past 7 days (Daily run at 00:30)
- ● Past 28 days (Daily run at 03:20)

Option #2: Cache a Bigger Result Set

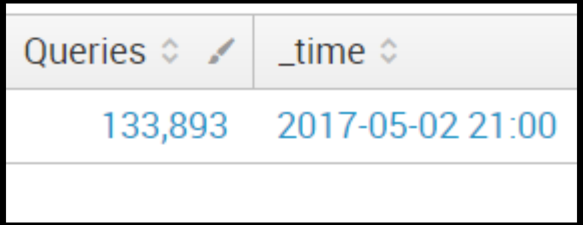
Sample SPL

```
| loadjob savedsearch= "myusername:search:My Saved Search"
```



Queries	_time
137,000	2017-04-30 21:00
174,329	2017-05-01 21:00
133,893	2017-05-02 21:00
137,947	2017-05-03 21:00
113,227	2017-05-04 21:00
180,698	2017-05-05 21:00
191,319	2017-05-06 21:00

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



Queries	_time
133,893	2017-05-02 21:00

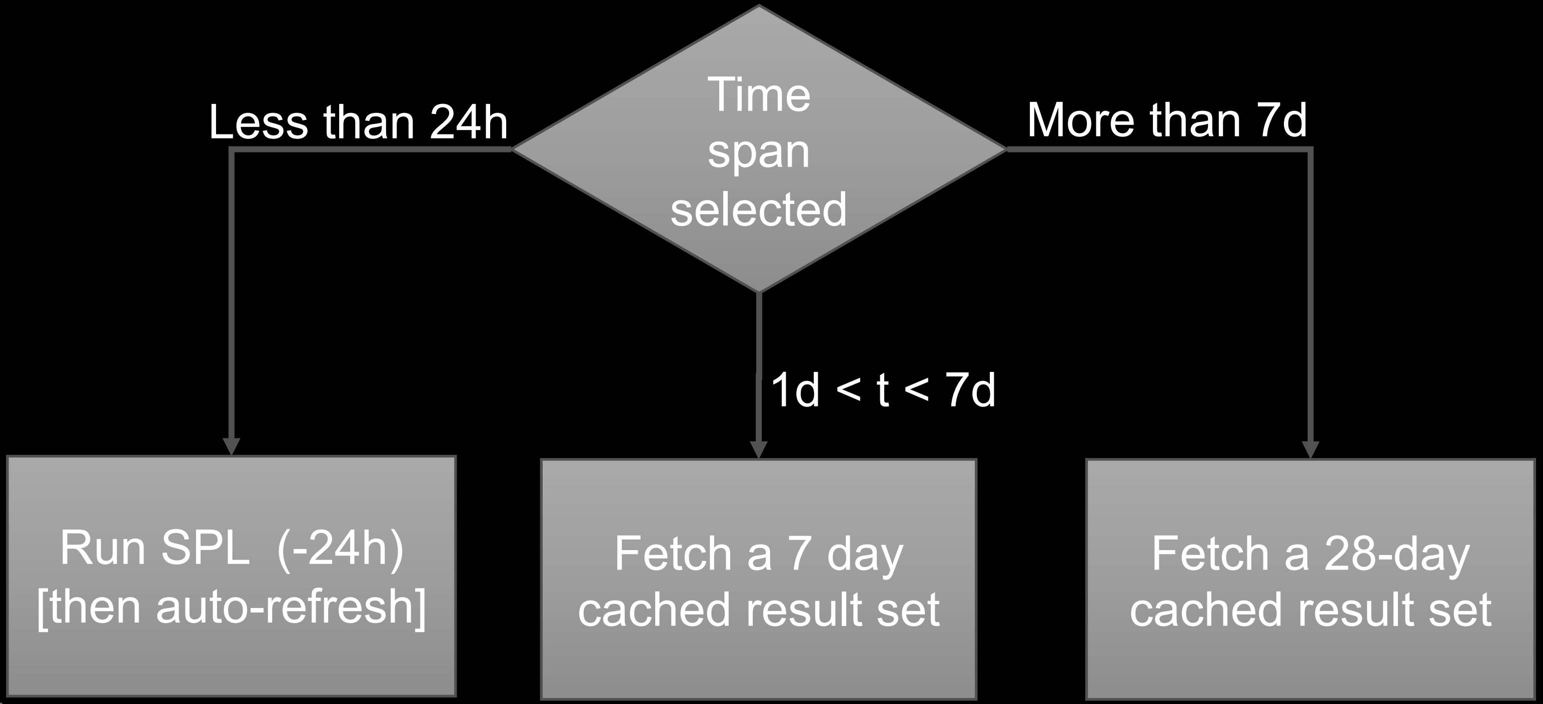
Splunk Documentation
http://docs.splunk.com/Documentation/Splunk/6.6.3/SearchReference/Where?r=searchtip
http://docs.splunk.com/Documentation/Splunk/6.6.3/SearchReference/DateandTimeFunctions#relative_time.28X.2CY.29

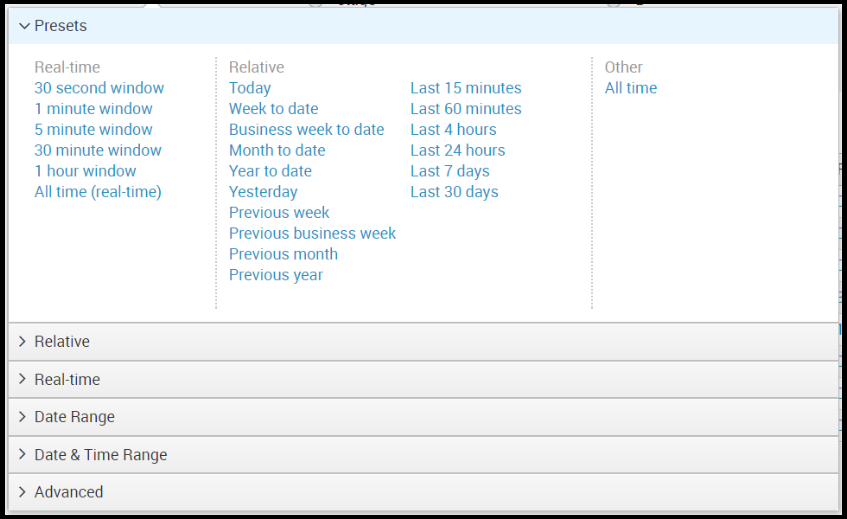
splunk> .conf2017

Option #3: Standard Time Picker Input

Use the standard time input, but test selected period

- Standard Splunk time input


Time span selected
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

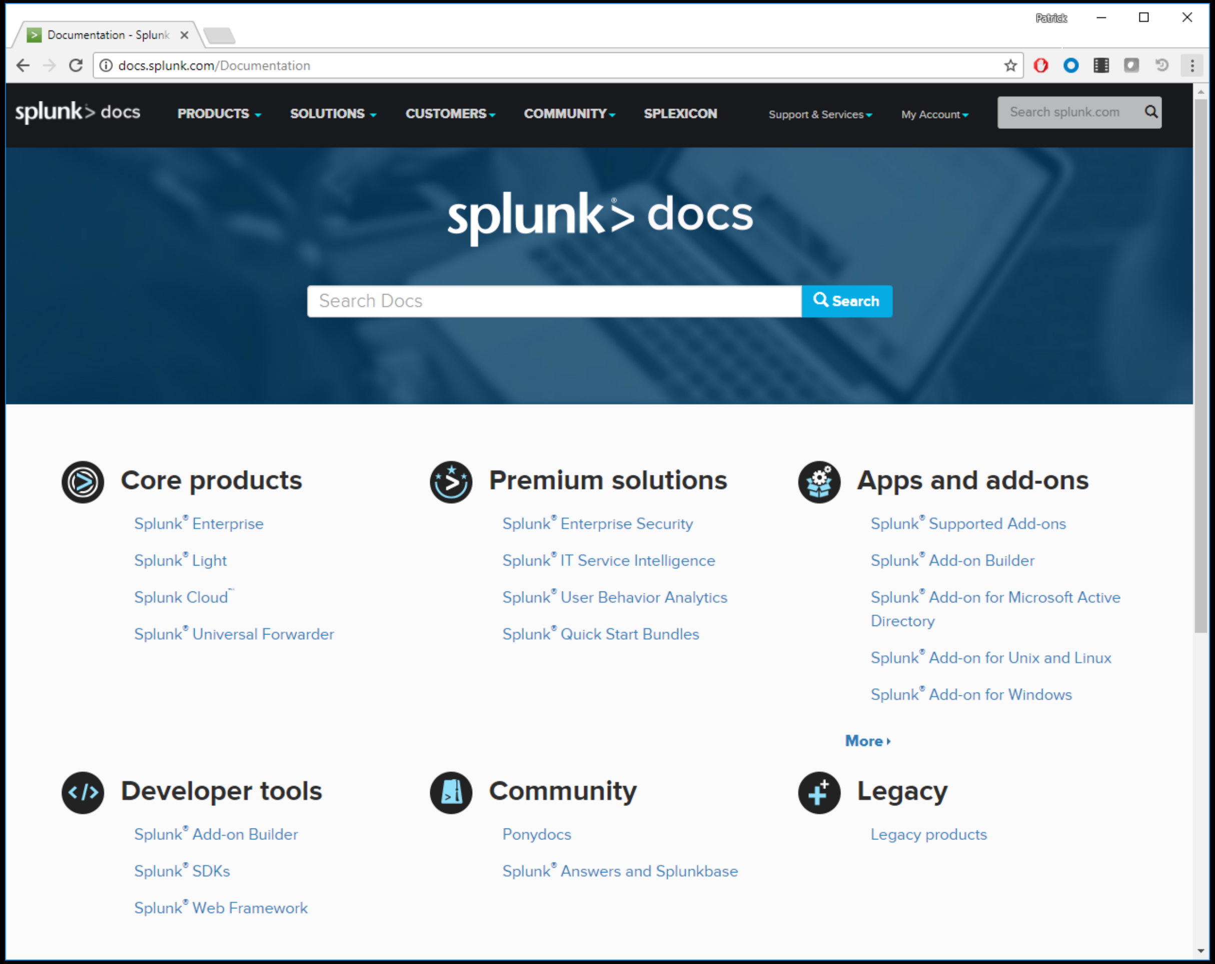
```
<change>
  <condition match="(relative_time(now(), $time_tok.latest$) -
                     relative_time(now(), $time_tok.earliest$))
                     &lt;= 86400">
    <!-- If selected time spans < 1d, run real-time query. -->
    <set token="less_than_1_day">true</set>
    <set token="short-ish">true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



Core products
Splunk® Enterprise
Splunk® Light
Splunk Cloud™
Splunk® Universal Forwarder

Premium solutions
Splunk® Enterprise Security
Splunk® IT Service Intelligence
Splunk® User Behavior Analytics
Splunk® Quick Start Bundles

Apps and add-ons
Splunk® Supported Add-ons
Splunk® Add-on Builder
Splunk® Add-on for Microsoft Active Directory
Splunk® Add-on for Unix and Linux
Splunk® Add-on for Windows
More ›

Developer tools
Splunk® Add-on Builder
Splunk® SDKs
Splunk® Web Framework

Community
Ponydocs
Splunk® Answers and Splunkbase

Legacy
Legacy products

Key Takeaways

You have options…

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

splunk> .conf2017