Data extraction with SignalFlow

You want to extract past or streaming time series data that has been sent to Splunk Infrastructure Monitoring. You want to extract “raw” data (that is, the metrics and their values), as well as data that has been processed by Splunk analytics.

Procedure

Splunk Infrastructure Monitoring provides a language called SignalFlow that is primarily used to describe computations for Splunk's real-time analytics engine. The SignalFlow command-line interface (CLI) for the Splunk v2 API outputs historical or streaming data in text format to a live feed, to a simple graphical display, or as CSV-formatted text. It is good to use to:

- Export streaming data
- Export data with a relative time range (e.g. last 15 minutes)
- Export raw data (no analytics applied), for a specific past time range, using a default rollup and resolution
- Export raw data (no analytics applied), for a specific past time range, at a rollup or resolution different from the Splunk defaults
- Export data with analytics applied in a way that isn’t reflected in a chart

To use SignalFlow for data extraction you must become familiar with either the Splunk v2 API (to use it with “curl” or via a client library) or the CLI. You need to learn which options provide the information you need and how to build the query using the API or the CLI. This includes an understanding of maxDelay, rollups, and resolutions.

The advantages of SignalFlow are:

- It provides powerful capabilities that let you filter data, apply analytics, and specify options for resolution, rollup, and other advanced settings.
- You can export streaming data, meaning you can stream data directly to another target as it is sent to Splunk Infrastructure Monitoring.
- You can specify relative time ranges, such as the last 15 minutes, or from 2 days ago to 1 day ago, rather than only using milliseconds since epoch.

The SignalFlow CLI is not an officially supported tool. It is intended to be an example of how to use the SignalFlow analytics language part of the signalfx-python library.

When you invoke SignalFlow, you will see the prompt ->. You can then enter a SignalFlow program (even across multiple lines) and press "" to execute the program and visualize the results. Press ^C at any time to interrupt the stream, and again to exit the client. To actually extract data, you use the "publish()" API.
Example usage

In this example, we stream live data directly to the screen.

```
$ signalflow
   -> data('jvm.cpu.load').mean(by='aws_availability_zone').publish()
```

To see current parameter settings, use the . command (press ")

```
-> .
   {'max_delay': None,
    'output': 'live',
    'resolution': None,
    'start': '-1m',
    'stop': None}
->
```

To set a parameter, use ".<parameter><value> ":

```
-> .start -15m.
   -> .stop -1m
   -> .
   {'max_delay': None,
    'output': 'live',
    'resolution': None,
    'start': '-15m',
    'stop': '-1m'}
```

Use the commands in a program named program.txt to:

1. extract non-streaming data from 15 minutes ago to 1 minute ago
2. output it in CSV format to a file named csv-to-plot.csv

```
$ signalflow --start=-15m --stop=-1m --output=csv < program.txt | csv-to-plot
```

Troubleshoot SignalFlow

When you use SignalFlow, the data is processed using the full capabilities of the Splunk analytics engine, which includes special handling of jitter and lag in data arrival times. There are two reasons that the analytics engine is waiting to process the computation.

The first is "max_delay", which is the amount of time we wait for delayed data before processing analytics. If not specified or set to None, the value of "max_delay" is determined automatically, based on Splunk's analysis of incoming data. To avoid delays in getting data from SignalFlow, set the "max_delay" parameter to 1s. This means that even if data is delayed, Splunk Infrastructure Monitoring will process the analytics after 1 second, without the missing data.

The information provided in Splunk Lantern is intended for informational and educational purposes only. All information is provided in good faith, however, Splunk disclaims any and all representations and warranties, express and implied, regarding the information provided, including without limitation any warranties and representations regarding the completeness, adequacy or accuracy of the information. You agree to take full responsibility for the results arising from the use of the information provided.
信号流

$ signalflow
  -> .max_delay 1s

如果你想将“max_delay”设置为更长的时间，确保你的“stop”值是一个时间，比“max_delay”更大。例如，如果你想要一个“max_delay”为30秒，那么使用一个“stop”值为-31秒或更早。

  -> .max_delay 30s
  -> .stop -31s

第二个可能导致计算延迟的原因是工作负载的解决。SignalFlow必须等到当前解决窗口的末尾，然后再进行计算。例如，如果工作负载的解决时间是300000（5分钟）并且“stop”值是“None”（或未指定），SignalFlow将等待直到它拥有当前5分钟时间窗口的所有数据点，然后再进行任何计算。

为了避免延迟，确保你的“stop”值是一个时间，比现在的时间更大。例如，如果你在查看几个月前的数据，解决时间可能为3600000（1小时）。在这种情况下，使用一个“stop”值为-1h或更多。

  -> .stop -1h

如果最近的数据请求返回的数据过时一分钟，这个问题也可能与最大延迟有关。相反地，而不是使用“stop”值为“None”（或不指定），将“stop”值设置为-1m。

  -> .stop -1m

**Next steps**

你可能对与提取数据和监控相关的其他过程感兴趣。

[Extracting data from Splunk Infrastructure Monitoring use case.](#)