痒いところに手が届く
ネットワークリソースの視覚化

2016/4/15
JANOG37.5 第二部

富士通株式会社
岩田 浩真

自己紹介

@kooshin（岩田 浩真）
AS2510（富士通）の近くで閉域網を担当

ネットワークSE
ルータやスイッチでMPLS網の運用

Qiitaで記事投稿してます
ズンドコキヨシ with ルータ
- Cisco VIRLとMPLS-TEでズンドコルータを実現してみた
  http://qiita.com/kooshin/items/2e00cdeb53cf9cf4c51d

手の届かない痒いところ

痒いところ

ルータのリソース枯渇系のバグを踏む

ルータのリソースの監視が必要になる

手の届かないところ

ルータに監視用のSNMP MIBが無い

SNMPが届かない・許可されてない

痒いところに手が届く”孫の手”が必要

”孫の手”

手の届かないところへ

手段は選べないので割り切る

定期的にSSH/TELNETでshowコマンド実行

孫の手のカタチ

伝統的なCSV+Excel、RRDtool、SQLよりも

開発しやすくモダンで柔軟性が高い手段で実現したい

方法

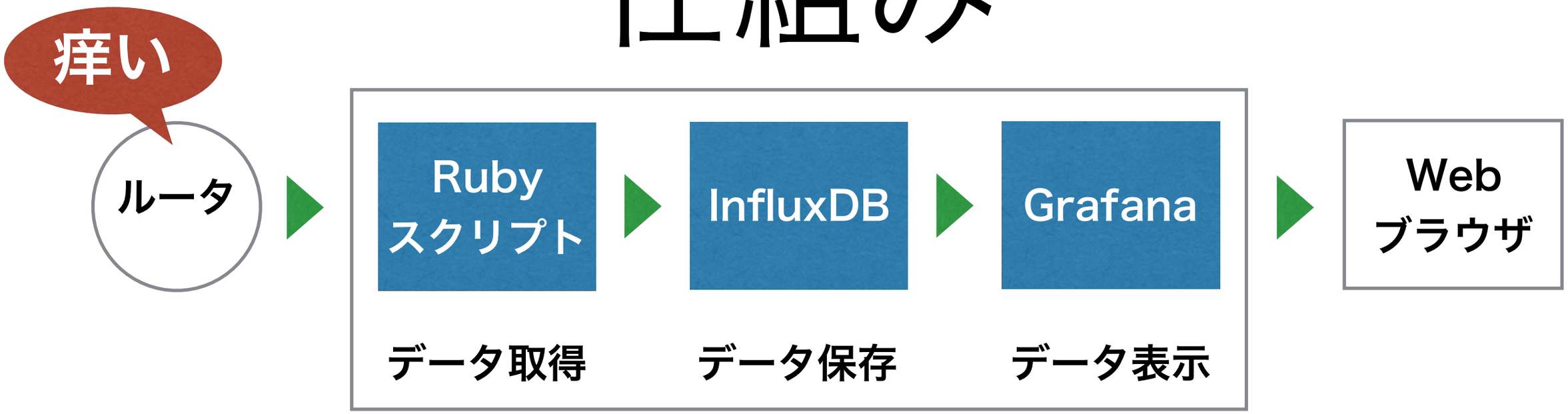
データ取得

自動化のため「**Rubyスクリプト**」から
SSH/TELNETして、showコマンドから
正規表現でデータを抽出

データ保存・表示

柔軟に取得項目を増やしたりしたいため、
時系列データベース「**InfluxDB**」と
かっこよく視覚化できる「**Grafana**」で試作

仕組み



処理の流れ

RubyスクリプトからSSHでルータにログインして、
showコマンドを実行して、
実行結果から正規表現でデータ抽出して、
InfluxDBにデータを時系列に保存して、
Grafanaでネットワークリソースを視覚化

Rubyライブラリ

expect4r

IOS/IOS XR/JUNOSルータへ簡単にTELNET/SSH

```
# Ciscoルータにログインするサンプル

require ‘expect4r’
ios = Expect4r::Ios.new_ssh(
  host: '192.168.0.1', user: ‘cisco’,
  pwd: 'cisco', enable_password: ‘cisco’
)

# コマンドshow interfaceを実行

result = ios.exec(‘show interface’).join.delete("¥r")

# あとは実行結果resultから正規表現でデータ抽出し、データ保存
```

正規表現

showコマンドの実行結果

```
iosv-1#show interfaces
GigabitEthernet0/0 is up, line protocol is up
  Hardware is iGbE, address is fa16.3efc.fae1 (bia fa16.3efc.fae1)
  Description: OOB Management
  Internet address is 172.16.1.217/24
MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
   reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
Full Duplex, Auto Speed, link type is auto, media type is RJ45
output flow-control is unsupported, input flow-control is unsupported
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:05:06, output 00:00:03, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
   3 packets input, 180 bytes, 0 no buffer
   Received 0 broadcasts (0 IP multicasts)
   0 runts, 0 giants, 0 throttles
   0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
   0 watchdog, 0 multicast, 0 pause input
   36 packets output, 2177 bytes, 0 underruns
   0 output errors, 0 collisions, 3 interface resets
   0 unknown protocol drops
   0 babbles, 0 late collision, 0 deferred
   1 lost carrier, 0 no carrier, 0 pause output
   0 output buffer failures, 0 output buffers swapped out
```

データ抽出の正規表現

```
REGEX_SHOWINT = %r{
  ^(?<interface>[\w\.\/]+)\ is
  .+?
  Total\ output\ drops:\ (?<out_drops>\d+)
  .+?
  (?<in_packets>\d+)\ packets\ input,
  \ (?<in_bytes>\d+)\ bytes
  .+?
  (?<in_errors>\d+)\ input\ errors,
  .+?
  (?<out_packets>\d+)\ packets\ output,
  \ (?<out_bytes>\d+)\ bytes
  .*?
  (?<out_errors>\d+)\ output\ errors,
  .+?
  (?<lost_carriers>\d+)\ lost\ carrier
}xm
```

InfluxDB + Grafana

時系列データベースInfluxDB

スキーマレスのため複雑なデータ構造でも保存可能

APIとSQLライクな操作をサポート

視覚化ツールGrafana

データをかっこよく視覚化

ダッシュボードやグラフをWebから作成

デモ環境

トラフィックを視覚化

10秒間隔でshow interfaceを実行

Cisco VIRLで2台の仮想ルータを準備

iperfでトラフィック生成

デモ画面


Traffic
Zoom Out
Last 1 hour
port: iosv-1 GigabitEthernet0/2
IN
104.89 kbps
OUT
2.05 Mbps
Traffic
2.5 Mpps
2.0 Mpps
1.5 Mpps
1.0 Mpps
500 kpps
0 pps
23:50
00:00
00:10
00:20
00:30
00:40
in_bits Max: 122 kpps Current: 105 kpps
out_bits Max: 2.209 Mpps Current: 2.054 Mpps
Packets
200 pps
150 pps
100 pps
50 pps
0 pps
in_packets Current: 165 pps
out_packets Current: 170 pps
Errors
1.00 pps
0.75 pps
0.50 pps
0.25 pps
0 pps
in_errors Max: 0 pps Current: 0 pps
out_errors Max: 0 pps Current: 0 pps
Lost Carriers
1.00
0.75
0.50
0.25
0
lost_carriers Max: 0 Current: 0
Drops
80
60
40
20
0
out_drops Max: 66 Current: 43

Qiitaで公開中

GrafanaとInfluxDBでネットワークリソースの視覚化

38 ストック　0 コメント

grafana 38　influxdb 41　Network 229　Cisco 58　Ruby 9928

kooshinが2016/04/10に投稿(2016/04/12に編集)・編集履歴(6)・問題がある投稿を報告する

ストック

Rubyスクリプトで、CiscoルータにSSHでログインして、CLIから10秒間隔でshow interface叩いて、実行結果から必要な値を正規表現で抜き出して、InfluxDBに保存して、Grafanaで視覚化してみました。

GrafanaとInfluxDBでネットワークリソースを視覚化してみました。データベースはInfluxDB、ビジュアライゼーションツールはGrafana、データ取得はRubyスクリプト＋expect4rライブラリで実装しています。

本記事はUsing InfluxDB + Grafana to Display Network Statisticsを参考にしています。データ取得にInfluxsnmpを利用していますが、今回は自作のRubyスクリプトで実行しています。

Tweet B! 51 G+1 2
Like 2 Pocket 63

kooshin
159 Contribution
フォロー

人気の投稿

- GrafanaとInfluxDBでネットワークリソースの視覚化
- GoTTYでブラウザからルータを操作してみた
- ズンドコキヨシ with ルータ - Cisco VIRLとMPLS-TEでズンドコルータを実現してみた
- CiscoルータのポートをRuby/Sinatraで見える化して検証捗らせてみた
- ズンドコキヨシ with OpenFlow - RubyとTremaとOpen vSwitchでズンドコキヨシ変換プロキシを実装してみた

Qiitaで応援！新生活

http://qiita.com/kooshin/items/a770c6df33fe2ffb2d0b