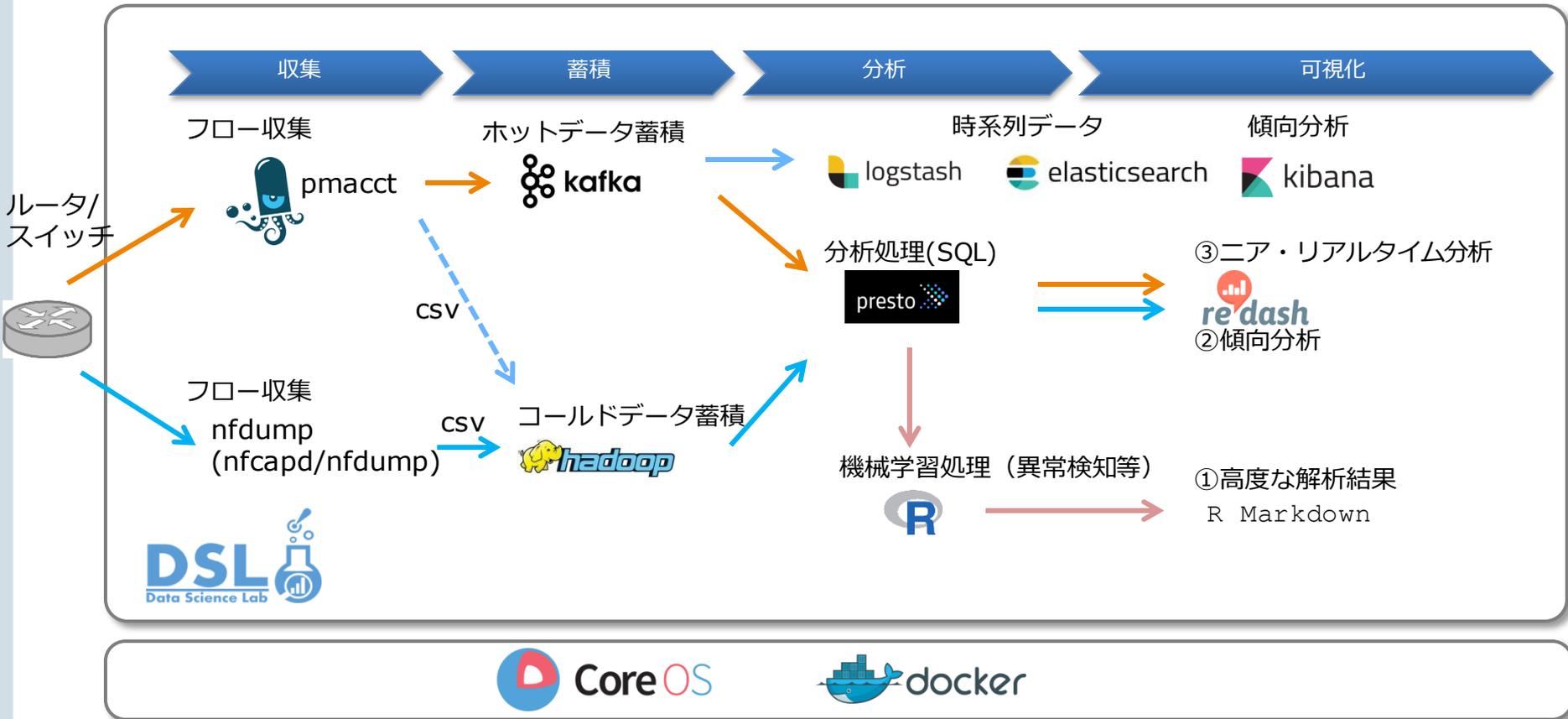


pmacct->kafka->presto->re:dash を使った高速なflow解析

JANOG39 Day1 BoF
2017年1月18日

西塚要
@__kaname__
kaname@nttv6.jp

flow解析の構成



flow収集部分

■ nfdump

- nfcapd: netflow(v1,v5/v7,v9,IPFIX)/sflow(v2/v4/v5)の収集を行い、n分ごとにファイルに出力する
- nfdump: nfcapdによって蓄積されたデータを読み出す→標準出力やファイル(csv形式など)に出力
- 信頼と実績！

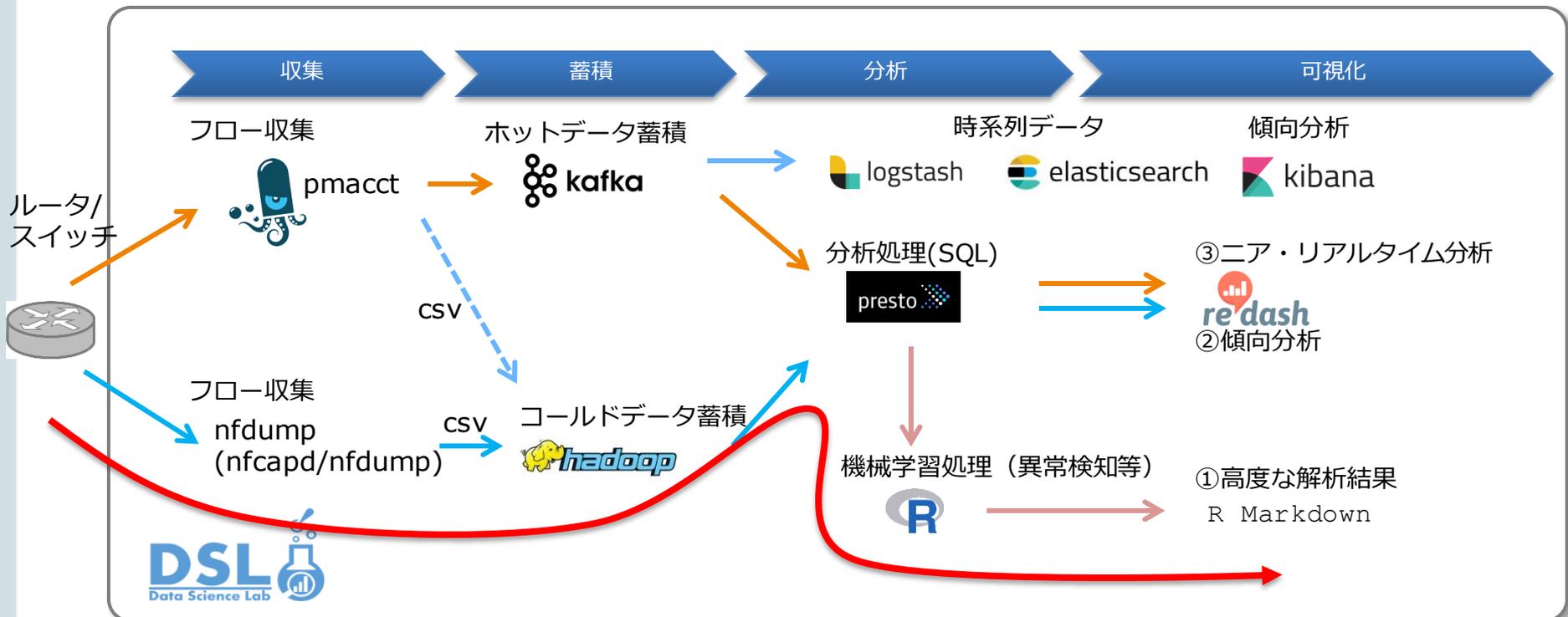
■ pmacct

- netflow(v5,v8,v9,IPFIX)/sflow(v2/v4/v5)の収集”も”行うことができる
- ファイル出力に加え、RDBMS, noSQL, Rabbit-MQ, kafkaとのコネクタがある
- paoloさんが頑張ってメンテしている！
 - ✓ [JANOG36:オープンソースのネットフローツールの運用](#)

事例1: 機械学習を用いた高度な解析

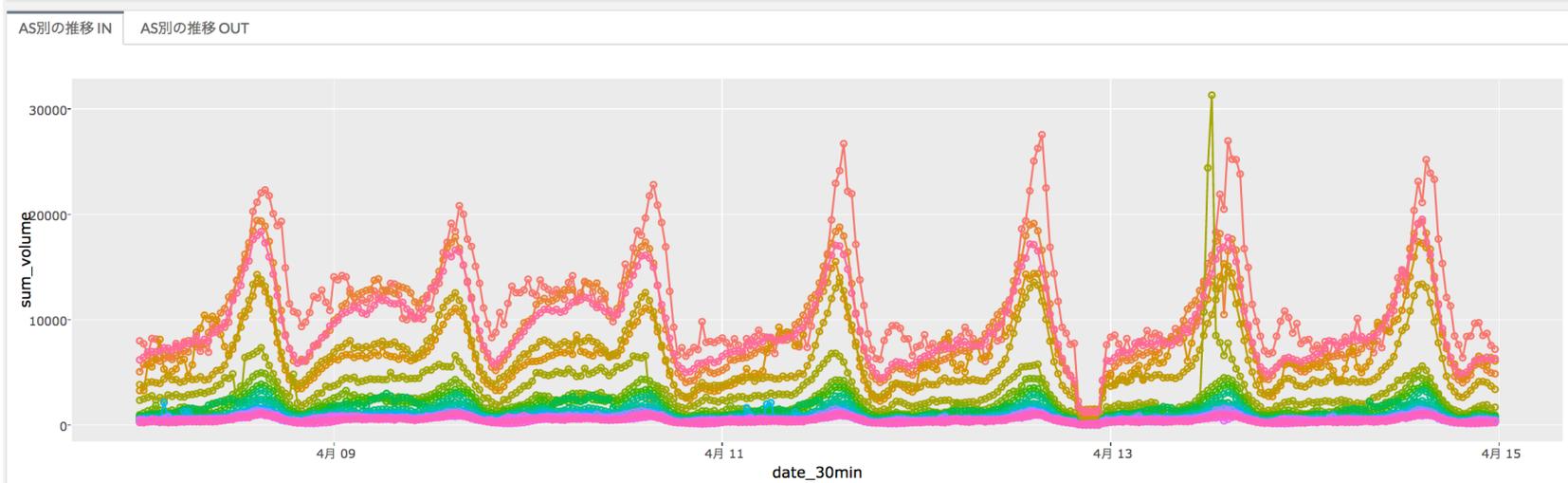
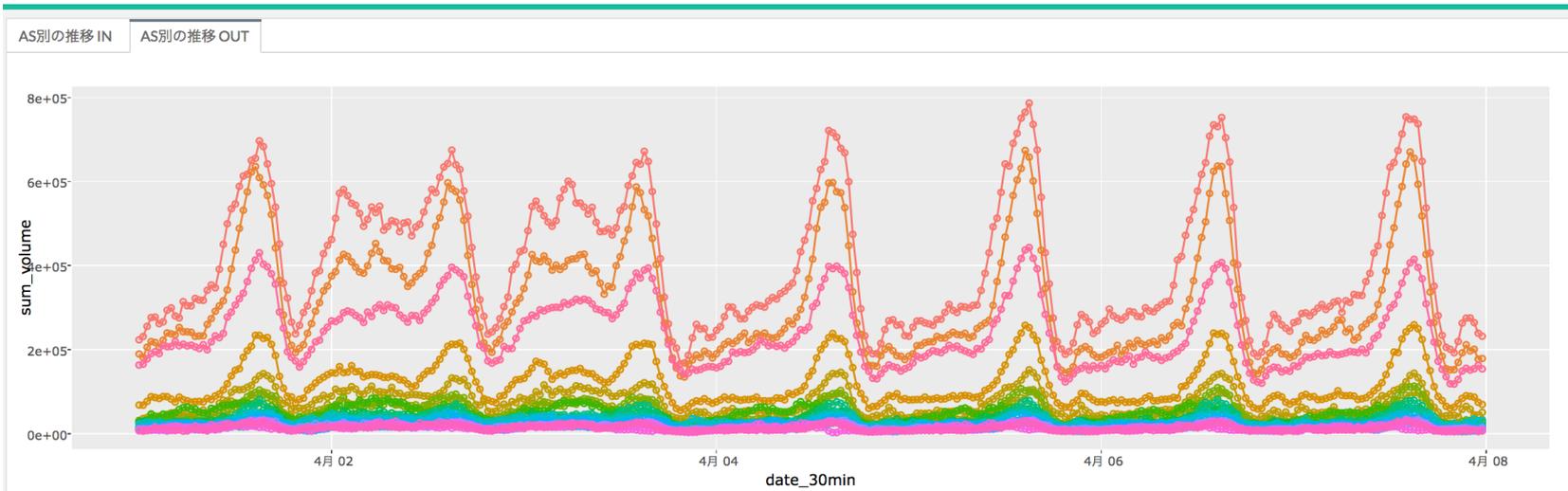
■ JANOG38:できる! データドリブン障害検知

- hadoop/hiveに蓄積された生フローデータをPrestoで時系列データに集約(早い!)
- 時系列データに対してRで機械学習処理



Rでのフロー解析

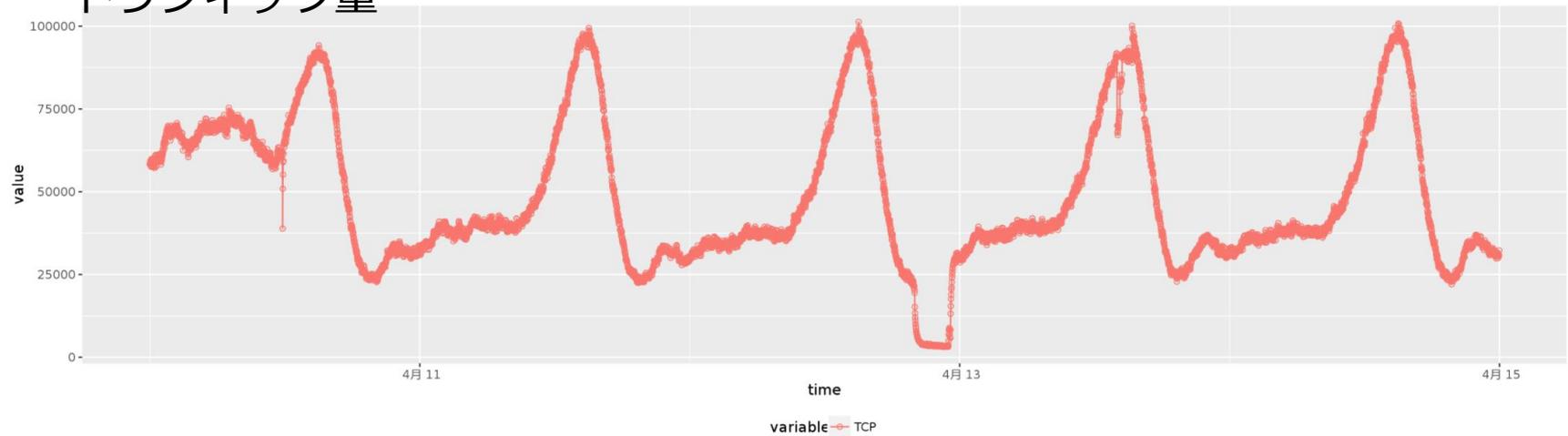
■ ASごとのトラフィックの推移



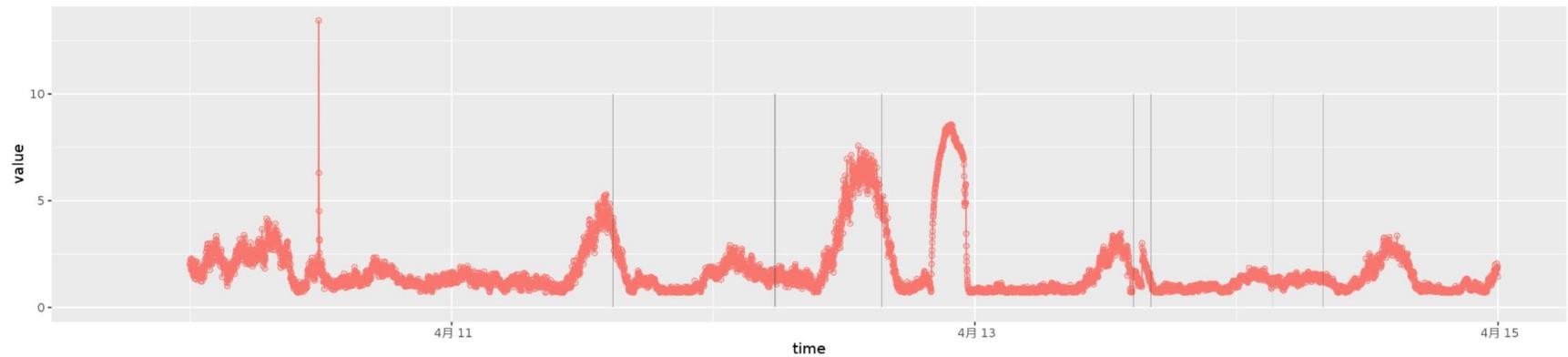
Rでのフロー解析+機械学習

■ トラフィック時系列への異常検知ロジックの適用

トラフィック量

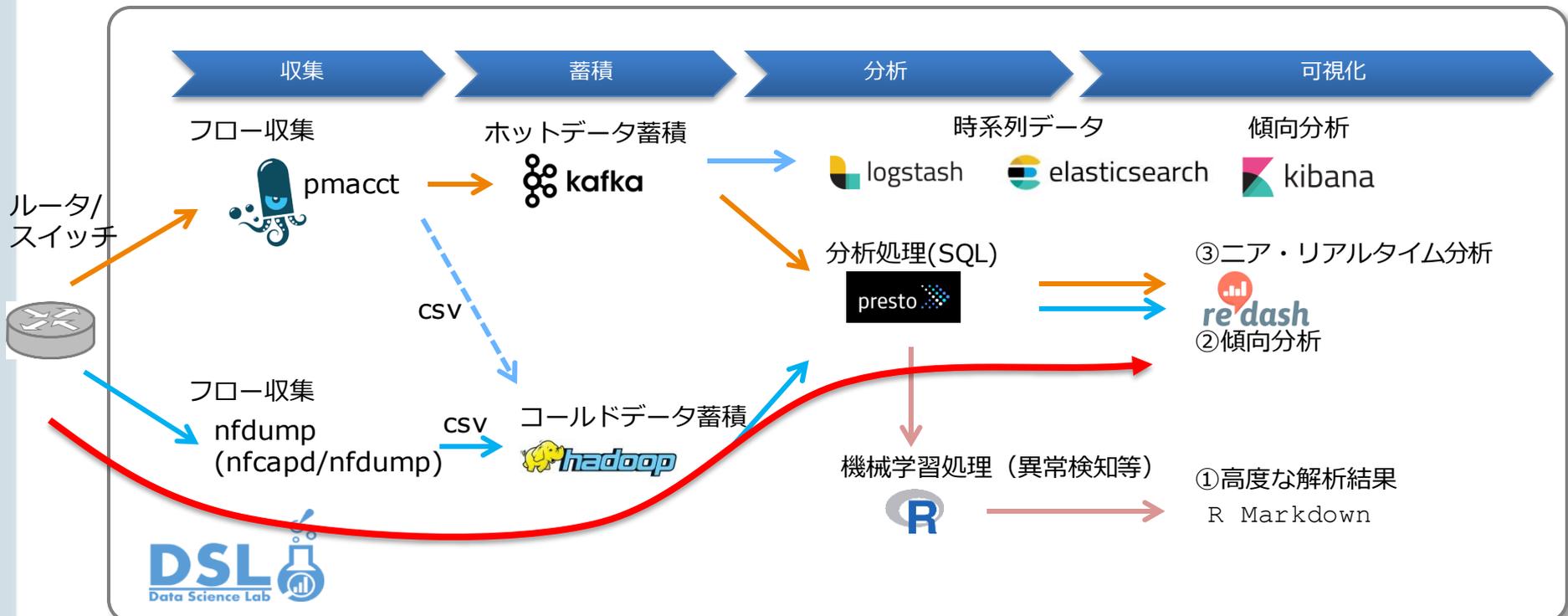


異常度



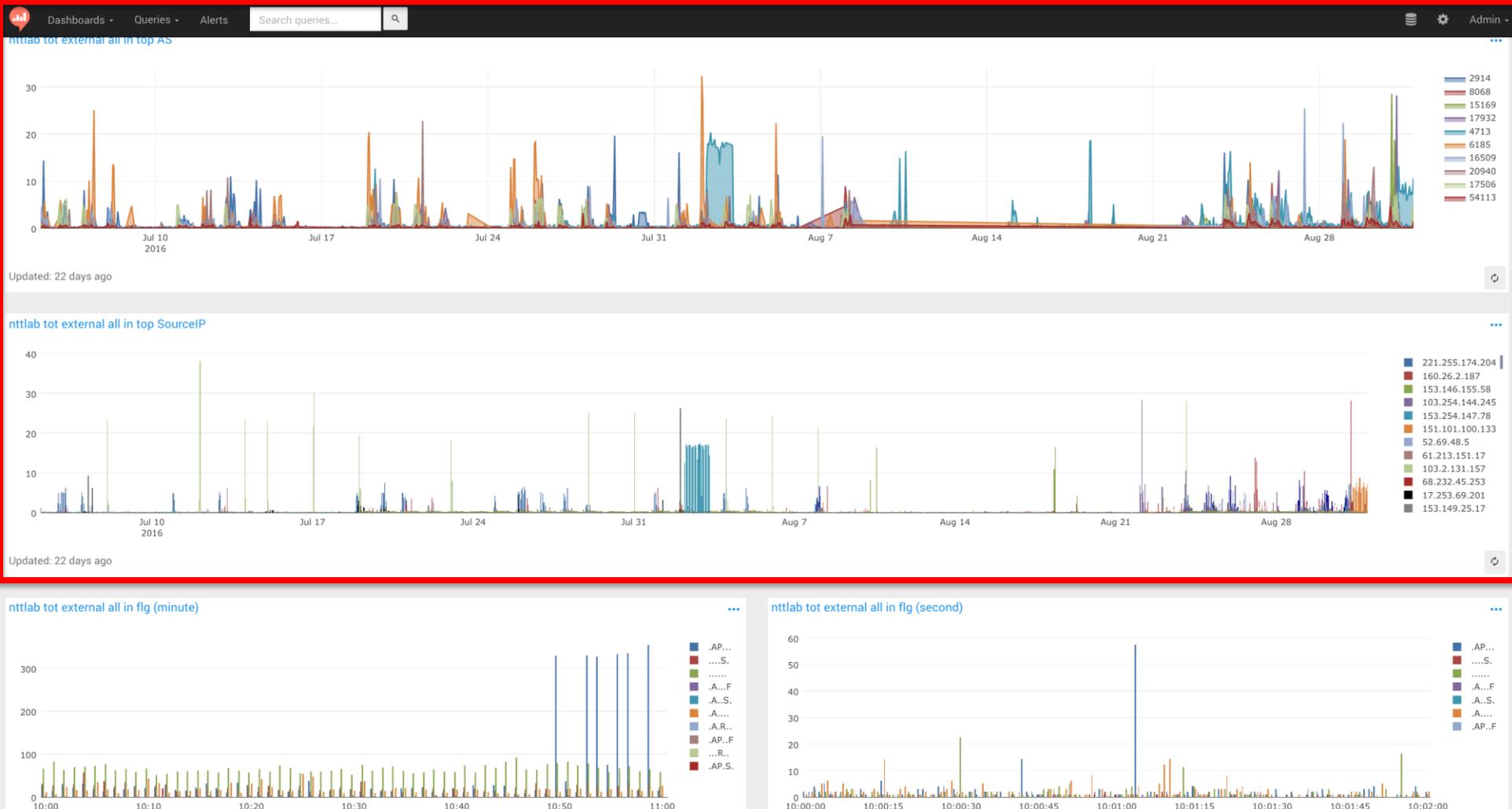
事例2: 高速な長期傾向分析

- モチベーション: 大量のフローデータに対して1ヶ月以上時間範囲で傾向分析をしたい
- hadoop/hiveに蓄積された生フローデータをPresto経由のre:dashで手早く可視化(早い!)



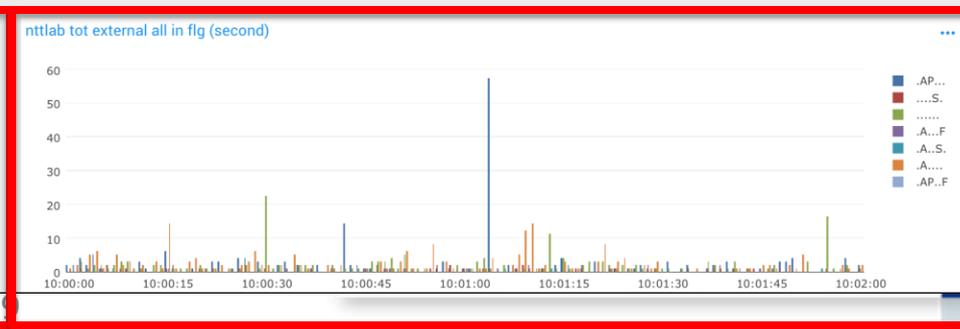
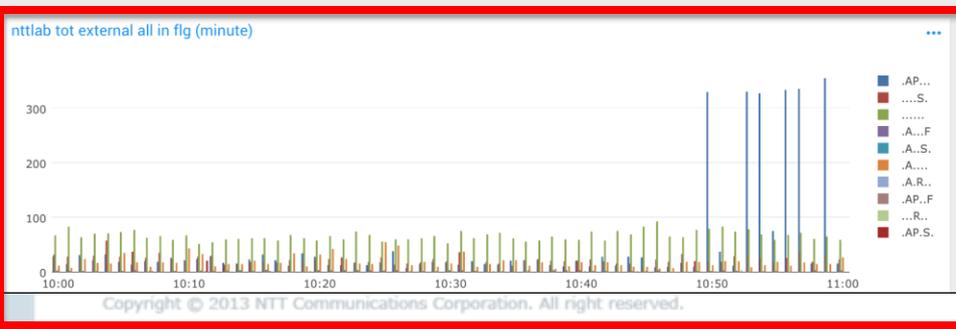
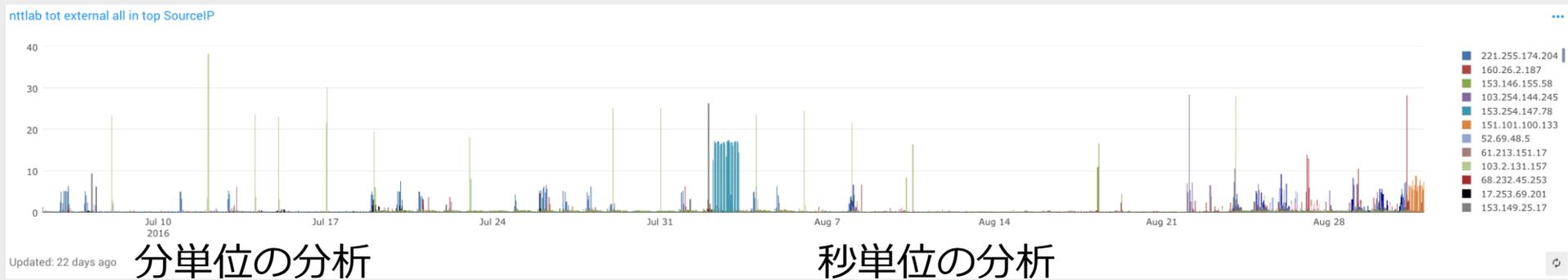
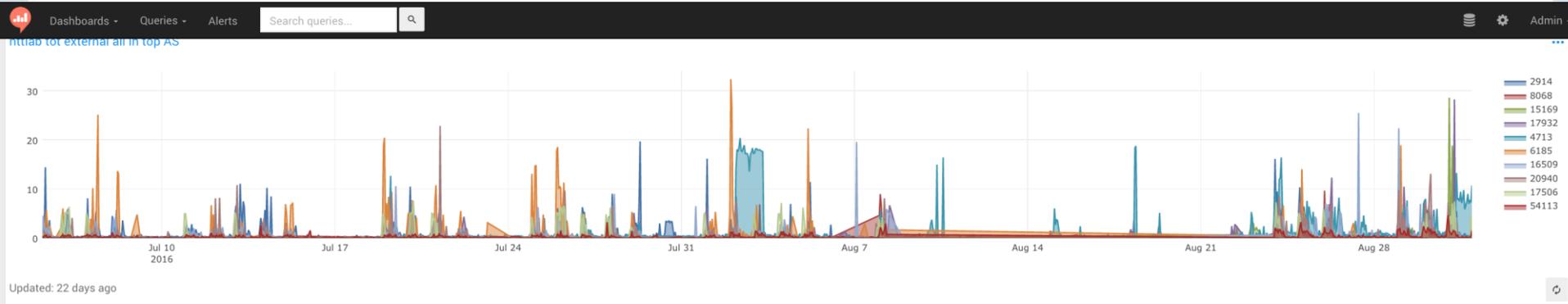
Prestoでの高速なクエリ実行

- TopN分析: 1ヶ月分のflowデータに対して、数秒で結果を返す



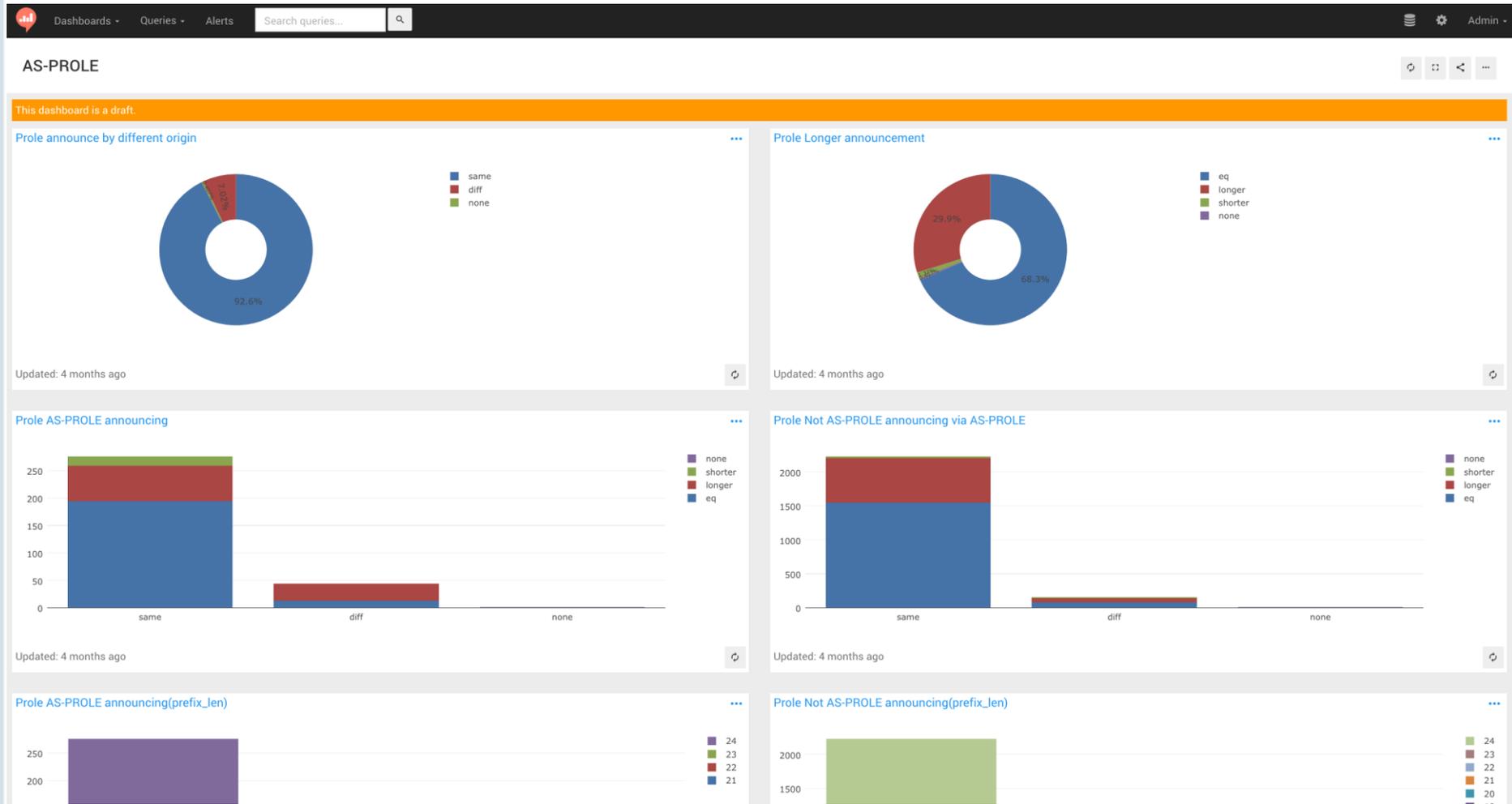
Prestoでの高速なクエリ実行

- どの期間に対しても、分単位や秒単位の分析が可能



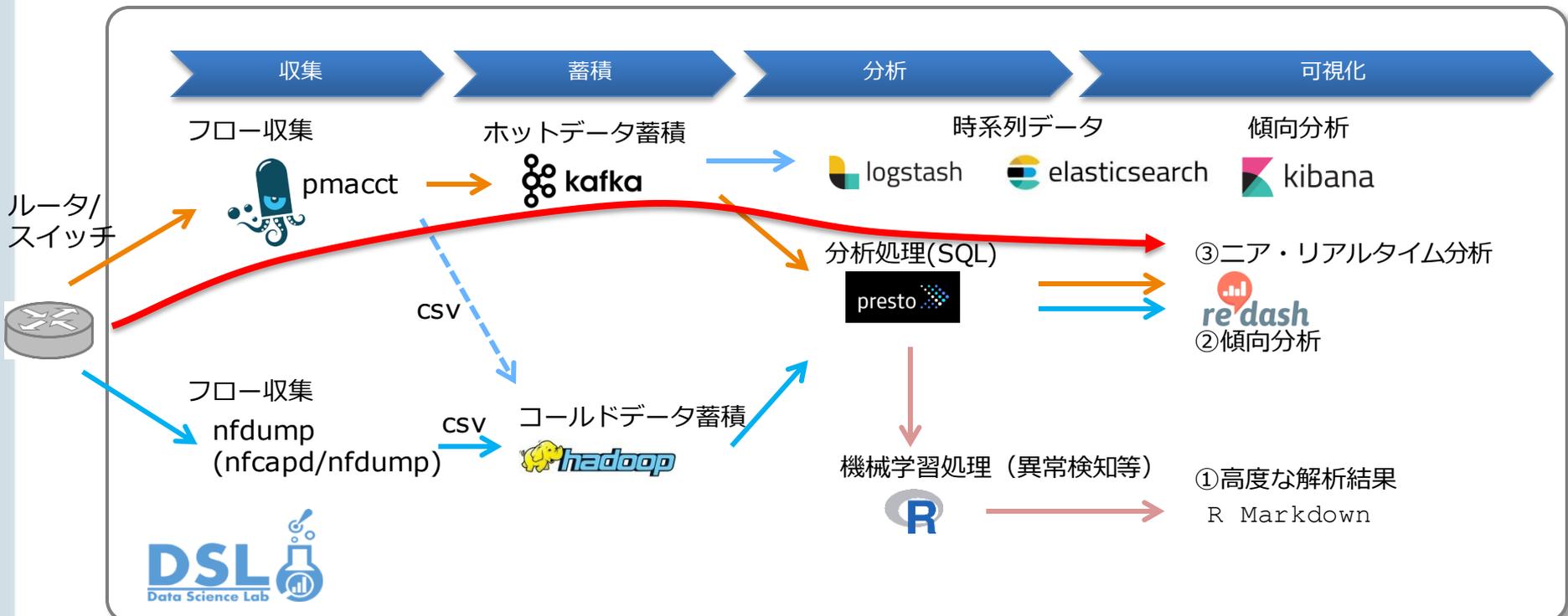
(参考) その他のデータとの連携

- re:dashには色々なデータを入れることができます
 - BGP解析の例: [IRS25 DDoS対策あれこれ](#)



事例3: 即時の短期傾向分析

- モチベーション: kafkaの即時性を利用して、1分未満のフロー解析を目指したい
 - 発生して1分未満でのDDoS攻撃検知など
- pmacct->kafka->presto->re:dash を試した



pmacct kafka-plugin

■ インストール

```
# 必要なパッケージのインストール
apt-get install libpcap-dev libjansson-dev
# 最新版の librdkafka のインストール
git clone https://github.com/edenhill/librdkafka.git; cd librdkafka; ./configure; make; make install
# kafka を利用するオプションをつけてpmacctのインストール
git clone https://github.com/pmacct/pmacct.git; cd pmacct; ./autogen.sh; ./configure --enable-kafka --enable-jansson --enable-ipv6; make; make install

# 確認
container:/# pmacctd -V
Promiscuous Mode Accounting Daemon, pmacctd 1.6.2-git (20170117-00)
'--enable-kafka' '--enable-jansson' '--enable-ipv6'
```

For suggestions, critics, bugs, contact me: Paolo Lucente <paolo@pmacct.net>.

■ ハマりポイント

- librdkafkaがインストールされたパスに注意
 - ✓ [QUICKSTART IX. Running the Kafka plugin](#) 参照

pmacct kafka-plugin

■ pmacct 設定

```
# nfacctd.conf
aggregate: src_mac, dst_mac, vlan, cos, etype, src_as, dst_as, peer_src_ip, peer_dst_ip, in_iface, out_iface,
src_host, src_net, dst_host, dst_net, src_mask, dst_mask, src_port, dst_port, tcpflags, proto, tos,
sampling_rate, timestamp_start, timestamp_end, timestamp_arrival
```

```
nfacctd_port: 2055
```

```
plugins: kafka
kafka_output: json
kafka_topic: pmacct-janog
kafka_refresh_time: 60
kafka_history: 1m
kafka_history_roundoff: m
kafka_broker_host: <kafka IP address>
kafka_broker_port: 9092
```

```
# 起動
```

```
nfacctd -f /etc/nfacctd.conf -dD
```

■ ハマリポイント

- nfacctd のデフォルトのnetflow待ち受けは 2100/udp
- kafkaのトピックに"."(ドット)を使うと、ここではOKだが、2ページ後のprestoの設定でハマリポイントあり

kafka-topic

■ kafka-topic が生成される

- Trifecta でデータが溜まり始めたことを確認

Trifecta v0.19.2

Inspect Observe Publish Query Decoders

Message Topics

17 of 17 Topic(s) shown [Hide empty]

ocnsearch-querylog (3,960,621)
pmacct-janog (419,047)
pmacct-test (1,487,277)
pmacct.acct
pmacct.flow (192,054,890)
pmacct.netflow
pmacct.nttwest
pmacct.sflow
pmacct.sflowtest
pmacct.test
probe-cloudwars
querylog
test-euler-topic4 (10,167,100)

Topic Offsets

Partition	Current Offset	Last Offset	Messages
0	419047	419047	419,047

View Key Message Offset 419047

```
{
  "event_type": "purge",
  "mac_src": "00:00:00:00:00:00",
  "mac_dst": "00:00:00:00:00:00",
  "vlan": 0,
  "cos": 0,
  "etype": "800",
  "as_src": 2914,
  "as_dst": 0,
  "peer_ip_src": "115.69.224.1",
  "peer_ip_dst": "0.0.0.0",
  "iface_in": 44,
  "iface_out": 95,
  "ip_src": "153.254.105.42",
  "net_src": "153.254.0.0",
  "ip_dst": "163.138.225.51",
  "net_dst": "163.138.225.48",
  "mask_src": 15,
  "mask_dst": 28,
  "port_src": 12346,
  "port_dst": 12446,
  "tcp_flags": "0",
  "ip_proto": "udp",
  "tos": 192,
  "sampling_rate": 0,
  "timestamp_start": "2017-01-17 12:55:45.0",
  "timestamp_end": "2017-01-17 12:55:45.0",
  "timestamp_arrival": "2017-01-17 12:56:00.613202",
}
```

presto

■ presto設定

kafka.propertiesの中にあるkafka.table-namesの値に、追加したいtopic名を記述

```
#<presto>/etc/catalog/kafka.properties
connector.name=kafka
kafka.nodes=<IP>:<port>, <IP>:<port>, <IP>:<port>
kafka.table-names=pmacct-janog
kafka.hide-internal-columns=false
```

テーブルの型定義をjsonファイルで記述

```
#<presto>/etc/kafka/pmacct-janog.json
```

```
{
  "tableName": "pmacct-janog",
  "topicName": "pmacct-janog",
  "dataFormat": "json",
  "message": {
    "dataFormat": "json",
    "fields": [
      {
        "name": "event_type",
        "mapping": "event_type",
        "type": "VARCHAR"
      }
    ]
  }
}
```

(snip)

■ ハマりポイント

- table名には"."が使えない(topic名に"."を使っていると、揃えようとしてやりがち)

re:dash で接続

- data sources に type=presto, catalog=kafka で追加

Dashboards ▾ Queries ▾ Alerts Search queries... 🔍

Settings

DATA SOURCES USERS GROUPS ALERT DESTINATIONS QUERY SNIPPETS

Name
presto-note01-kafka

Type
Presto

Catalog
kafka

Host
xxx

Port
18080

Schema

Username
redash

Save Delete Test Connection

re:dash で接続

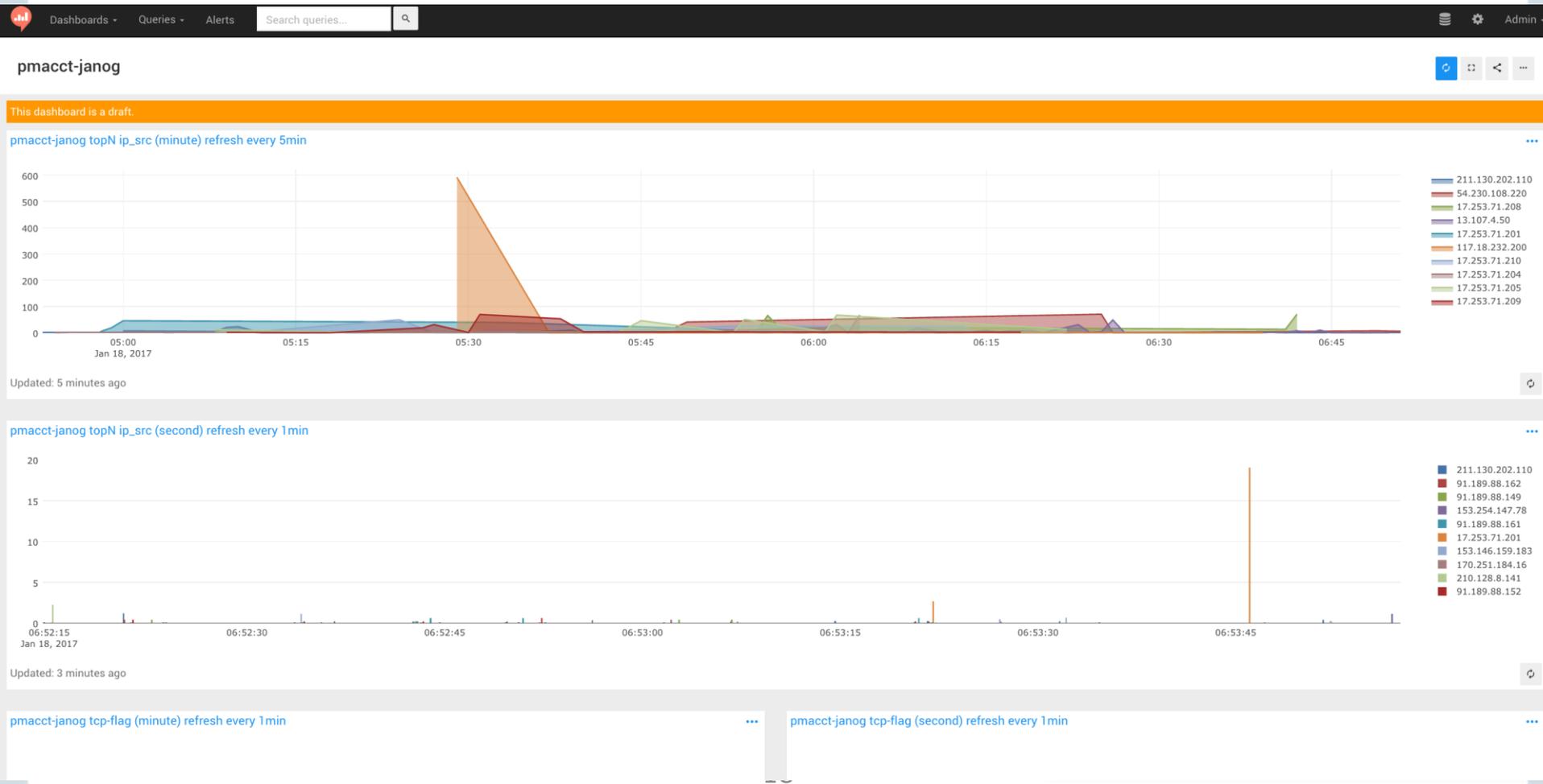
- 作成した data sources を指定して、select * from <table名> でデータが取れることを確認

The screenshot shows the re:dash interface for a query named "pmacct-janog SELECT *". The query is being executed against the "presto-note01-kafka" data source. The query text is "1 select * from 'pmacct-janog' limit 10". The interface includes a search schema panel on the left with a list of schemas: default.ocnsearch-querylog, default.pmacct-test, default.probe-cloudwars, default.tot_syslog-topic4, default.twitter-tweet, and default.visionalist. The query execution status shows "Runtime 2s", "Rows 10", and "Last update 16 hours ago". The table header at the bottom lists columns: KAFKA_KEY, EVENT_TYPE, MAC_SRC, MAC_DST, VLAN, COS, ETYPE, AS_SRC, AS_DST, PEER_IP_SRC, PEER_IP_DST, IFACE_IN, IFACE_OUT, IP_SRC, NET_SRC, IP_DST, NET_DST, MASK_SRC, MASK_DST, PORT_SRC, and PORT_DS.

結果：即時の短期傾向分析結果

■ 直近2分以内での秒単位のグラフ化

- 最新の時刻は、1分前くらい…(惜しい！)



最後に

- 一緒にトラフィック解析しませんか？
- ご連絡お待ちしております。
 - kaname@nttv6.jp