

RCP RestAPI

Tatiana Camelo
2015086298

1. Introdução

O trabalho a seguir se propõe em fazer sistema de consulta de banco de dados de interconexões. Através de conexão RCP, são trocados dados a cerca de Internet Exchange Points entre cliente e servidor.

RCP (remote procedure call) é uma forma de comunicação de dispositivos ligados a uma rede, se assemelhando a procedimento de chamadas locais, com a diferença que p RCP é feito através da rede e pode sofrer perda de dados ou intermitência.

APIs são interfaces de programação de aplicações para facilitar a interoperabilidade entre usuários e aplicações. Como o principal protocolo de comunicação de sistemas web 'e HTTP, uma forma de fazer acesso a APIs e por meio de uma requisição GET, POST ou DELETE HTTP e aplicando conceitos de arquitetura REST.

O sistema desenvolvido se trata de acesso e manipulação de dados em JSON de uma API REST via requisição GET HTTP, como será explicitado a seguir.

2. Setup

O sistema utiliza a biblioteca FLASK. Para utilizá-lo, é necessário instalar e criar a variável de ambiente.

```
$ export FLASK_APP=servidor.py  
$ python -m flask run
```

Então é necessário executar o servidor:

```
python server.py 8080 net.json ix.json netixlan.json
```

Sendo que <8080> representa a porta de conexão, <net.json> <ix.json> e <netixlan.json> representam os arquivos de entrada para NET, IX e NETIXLAN, respectivamente.

Esta mensagem irá aparecer, mostrando o IP local e a porta de conexão.

```
* Running on http://127.0.0.1:5000/
```

Com este endereço passa a ser possível fazer o acesso local e fazer cada requisição HTTP diretamente do navegador, por exemplo.

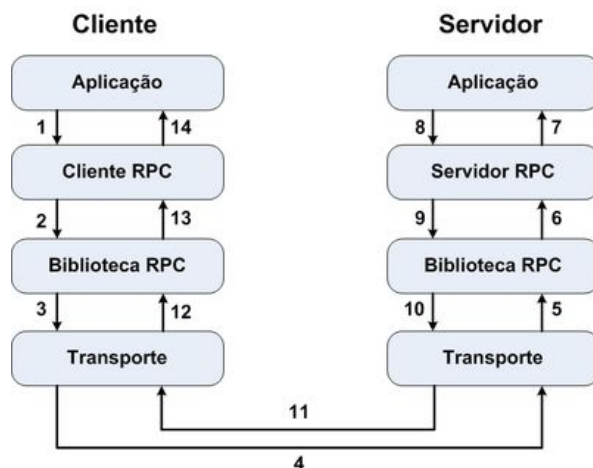
E o cliente:

```
python client.py 127.0.0.1:8080 1
```

Onde `<127.0.0.1>` é o IP e `<8080>` a porta para conexão. O último parâmetro `<1>` é o informativo da análise que deve ser feita, podendo ser a análise baseada em IXP por rede ou redes por IXP, sendo os parâmetros 0 e 1, respectivamente.

3. Modelagem

O cliente e servidor são interfaces para garantir comunicação e troca de dados entre dispositivos ligados a uma rede. O funcionamento utilizando RCP é explicitado abaixo:



O cliente faz conexão com os parâmetros de IP e PORT dados inicialmente e cria uma requisição GET de forma manual, sendo que cada requisição é uma nova conexão na rede. A requisição retorna valores da base acessada pelo servidor e são feitas análises sobre estes dados.

Para a análise 0, cria-se uma tabela TSV (tab separated value) contendo o identificador, nome da rede e número de IXPs associados ao cliente. Isso foi feito pela requisição do endpoint `('/api/ix')` comparado com os valores do endpoint `('/api/ixnets/<ix_id>')`. Vale ressaltar que para evitar sobrescrita da análise, foi feita verificação dos valores repetidos na análise, se forma a incrementar o valor se já existente ou torná-lo zero caso inexistente. Esta lista foi comparada com o endpoint `('/api/netname/<net_id>')` para acesso ao nome da rede.

Para a análise 1, também cria-se uma tabela TSV contendo identificador de IXP, nome de IXP e número de redes associadas. Isso foi feito através de

Já o servidor carrega as informações via arquivos de entrada no padrão JSON e exporta os dados com a requisição do cliente por três endpoints HTTP implementados via API Rest. O servidor apenas responde com os dados de cada requisição para que o cliente possa manipulá-las.

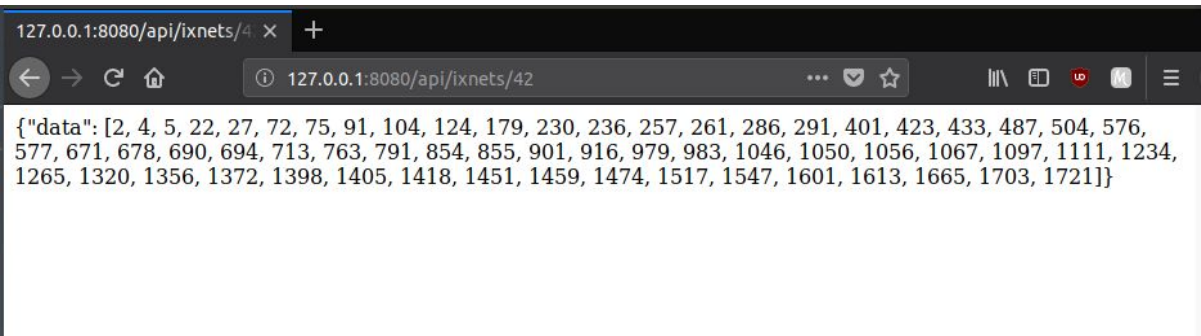
Os endpoints e exemplo das saídas originais são:

`/api/ix` - retorna todos IXPs



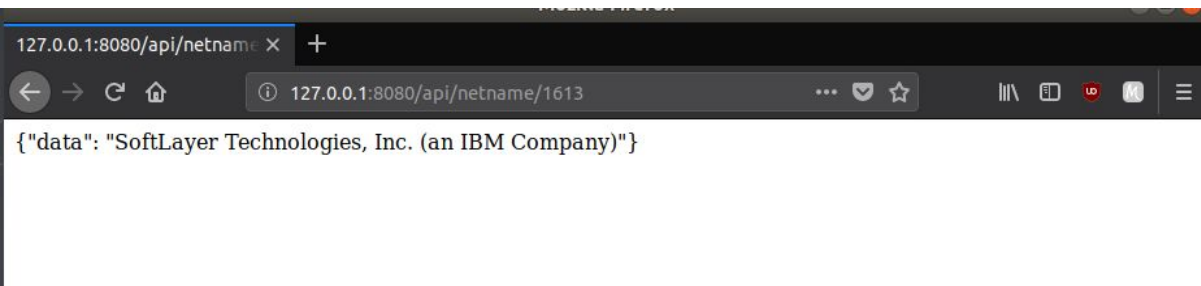
```
{
  "data": [
    {
      "proto_ipv6": true,
      "city": "Ashburn",
      "updated": "2016-11-23T21:40:34Z",
      "policy_email": "support@equinix.com",
      "name": "Equinix Ashburn",
      "tech_phone": "",
      "country": "US",
      "notes": "",
      "created": "2010-07-29T00:00:00Z",
      "org_id": 2,
      "proto_multicast": false,
      "region_continent": "North America",
      "website": "https://ix.equinix.com",
      "url_stats": "",
      "policy_phone": "",
      "status": "ok",
      "proto_unicast": true,
      "media": "Ethernet",
      "tech_email": "support@equinix.com",
      "id": 1,
      "name_long": "Equinix Ashburn Exchange"
    },
    {
      "proto_ipv6": true,
      "city": "Chicago",
      "updated": "2016-06-08T15:55:16Z",
      "policy_email": "support@equinix.com",
      "name": "Equinix Chicago",
      "tech_phone": "",
      "country": "US",
      "notes": "",
      "created": "2010-07-29T00:00:00Z",
      "org_id": 2,
      "proto_multicast": false,
      "region_continent": "North America",
      "website": "https://ix.equinix.com",
      "url_stats": "",
      "policy_phone": "",
      "status": "ok",
      "proto_unicast": true,
      "media": "Ethernet",
      "tech_email": "support@equinix.com",
      "id": 2,
      "name_long": "Equinix Chicago Exchange"
    }
  ]
}
```

`/api/ixnets/<ix_id>` - retorna lista de todos identificadores do IXP passado como parâmetro



```
{
  "data": [
    2, 4, 5, 22, 27, 72, 75, 91, 104, 124, 179, 230, 236, 257, 261, 286, 291, 401, 423, 433, 487, 504, 576, 577, 671, 678, 690, 694, 713, 763, 791, 854, 855, 901, 916, 979, 983, 1046, 1050, 1056, 1067, 1097, 1111, 1234, 1265, 1320, 1356, 1372, 1398, 1405, 1418, 1451, 1459, 1474, 1517, 1547, 1601, 1613, 1665, 1703, 1721
  ]
}
```

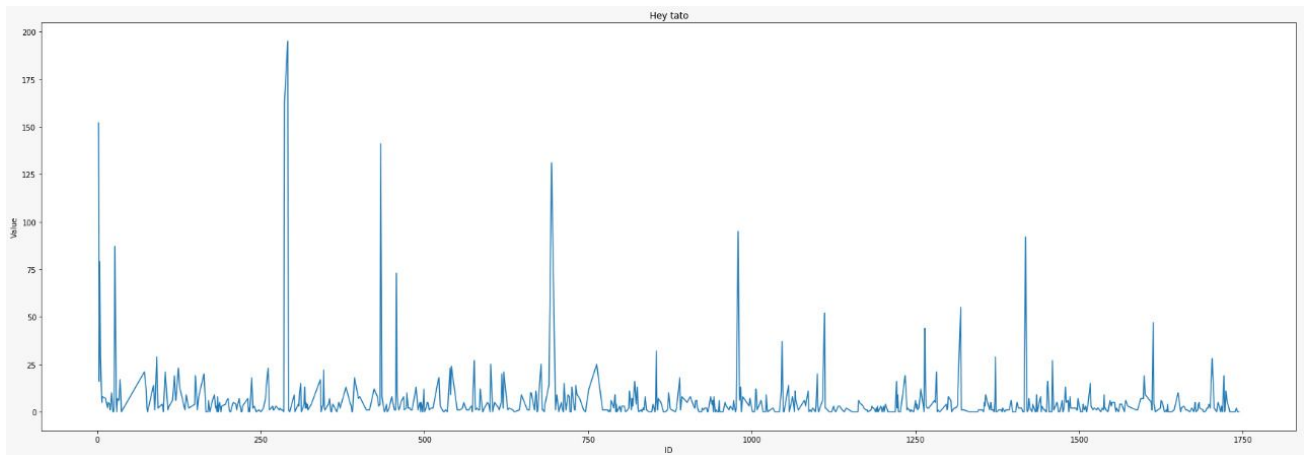
`/api/netname/<net_id>` - retorna o nome da rede identificado pelo id inserido no parâmetro



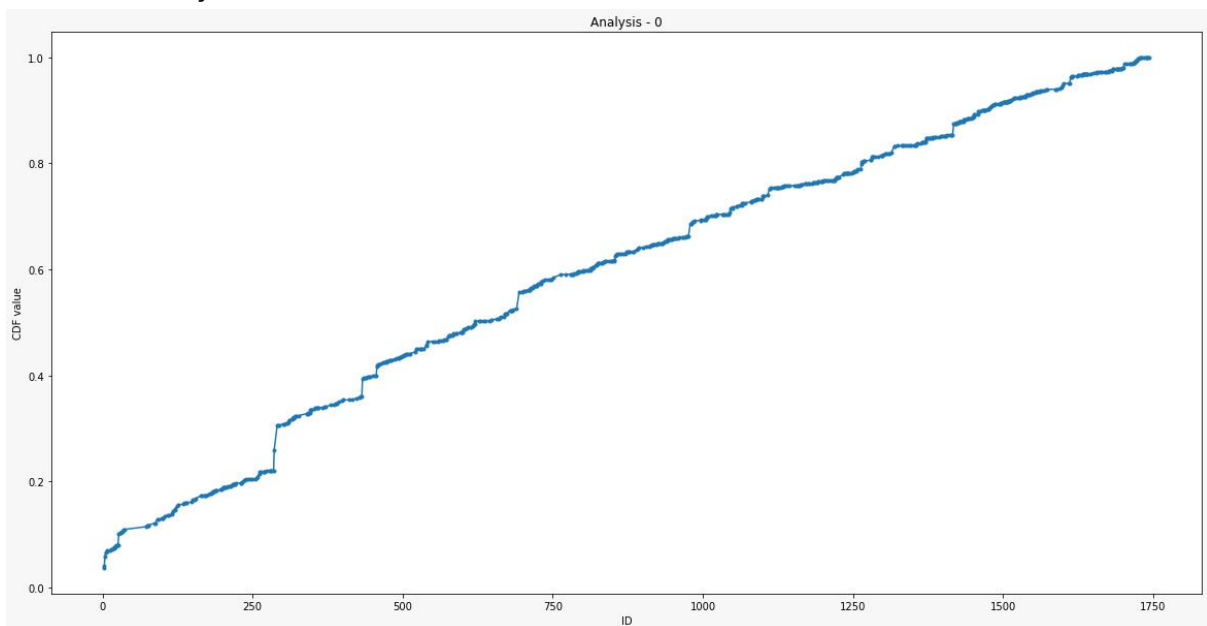
```
{
  "data": "SoftLayer Technologies, Inc. (an IBM Company)"
}
```

4. Análise

Análise 0 - IXP por redes

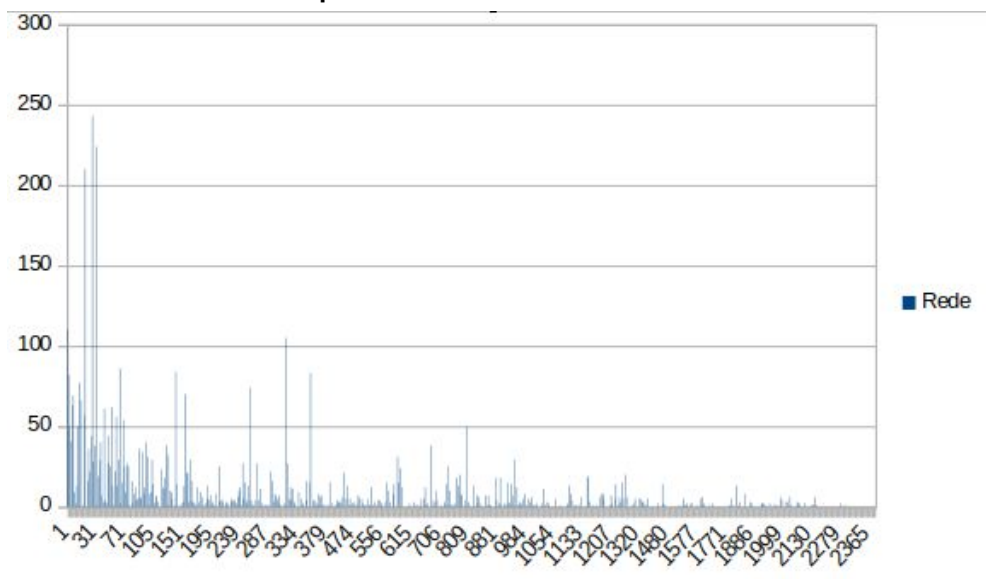


Distribuição direta

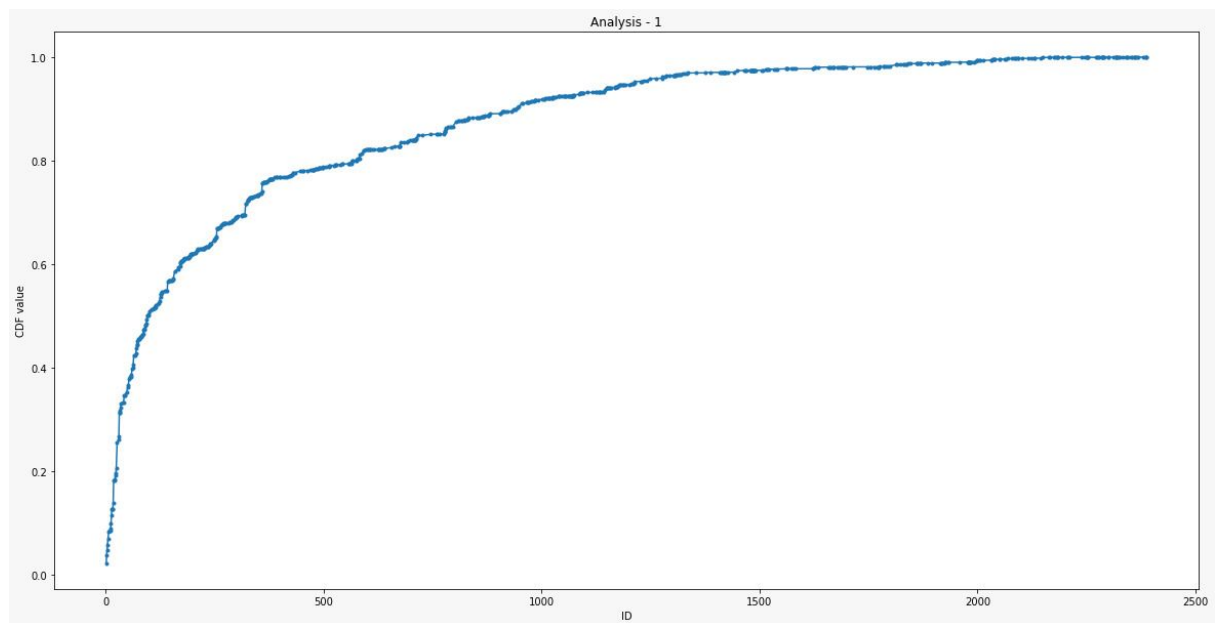


CDF (Cumulative distribution function)

Análise 1 - redes por IXP



Distribuição direta



CDF (Cumulative distribution function)

Pode-se entender que tem cerca de 80% dos IXP distribuídos para 20% das primeiras redes, então há um grande acúmulo inicial e então uma distribuição mais padronizada a partir disso. O gráfico da análise 0 mostra um crescimento quase linear, considerando a CDF, mostra distribuição proporcional entre os IXP por rede.

5. Conclusão

Um desafio deste trabalho foi a compreensão dos valores da análise para plotagem dos gráficos. Além de problemas técnicos, como dificuldade de exportação da saída final como arquivo devido ao encodamento inicial, o trabalho exaltou a importância da análise de requisições HTTP e seu tratamento, não apenas relativo aos arquivos JSON mas a atenção com os parâmetros de headers e tratamento específico do recebimento dos dados da chama retirando o header para tratar apenas os dados no body da resposta.

A análise dos dados também é desafiadora, já que requer bom entendimento e criação de hipóteses para compreensão dos dados. Entende-se que a distribuição de IXP é concentrada em pequeno número de redes e distribuída em pequena quantidade de redes. Além disso, a distribuição quase linear de IXP por redes mostra que há boa distribuição. Pode-se traçar paralelos com relação ao sistema atual de Internet, com acúmulos em servidores grades e conhecidos, o que causa instabilidade e sobrecarga do sistema.