

Redes de computadores - TP3 - API REST

Documentação e implementação

Guilherme Torres

December 2018

1 Introdução

O trabalho que se segue consistiu em construir um servidor que implementa uma API REST, responsável pela comunicação dos dados de um banco em formato JSON para um cliente, cujo papel é coletar estes dados e fazer algumas análises.

Os scripts foram escritos em Python 3 e podem ser executados da seguinte maneira:

```
./server port Netfile Ixfile Netixlanfile
```

e

```
./client IP:port Opt
```

como foi discutido na especificação.

2 Implementação

2.1 Servidor

O servidor tem uma implementação bastante simples. Ao início de sua execução, ele lê os dados do PeeringDB e guarda os dados em dicionários na RAM. As rotas retornam os dados do dicionário, segundo a especificação. Em seguida, ele roda o aplicativo usando Flask na porta especificada até ser desativado.

Não houve grandes dificuldades na implementação dessa parte do trabalho.

2.2 Cliente

O cliente possui dois modos de análises de dados: IXPs por redes e redes por IXPs. Ambos usam de um socket TCP que recebe uma header HTTP (exemplo: `'GET /api/ix HTTP/1.1 °: ' + args.ip_port + ' ° °'`) e é conectado ao servidor em seguida. A conexão deve ser renovada a cada requisição para evitar uma exceção do tipo BrokenPipe.

Para a análise do tipo zero (IXPs por redes), o processo leva um tempo para executar, devido ao número de requisições HTTP. Primeiramente é feita uma requisição para `/api/ix`, para obter os IDs dos IXPs, e a partir deles, são

feitas várias requisições para `/api/ixnets/id` para obter os IDs das redes que possuem associações com estes IXPs e, em seguida, dado um número n de IDs de redes, são feitas n requisições para `/api/netname/id` para identificar os nomes das redes que tem os IDs encontrados. Em seguida, o resultado é mostrado na saída padrão, no seguinte formato:

1412	Virgin Media	1	
1418	Amazon.com	1	
1510	Wi-Manx Limited		1
1723	Node4 Ltd	1	
261	Vodafone Global Network		1
286	Packet Clearing House	1	
291	Hurricane Electric	1	

Para a análise do tipo 1 (redes por IXPs), por sua vez, o cliente primeiramente faz uma requisição para a rota `/api/ix` e obtém os IDs dos IXPs que vai analisar e os seus nomes. A partir dos n ID que recebe, o cliente faz n requisições HTTP para o servidor para a rota `/api/ixnets/id` e conta o tamanho das respostas para cada ID, assim montando uma lista de resultados. Em seguida é imprimido na tela o resultado, no formato (exemplo):

1180	MegaIX Dallas	5
1183	6NGIX	0
1187	SYMC-IX Bangkok	0
1188	TC-IX	0
1191	SerinIX IX	0
1201	AIXP	1
1207	IX-Denver	8
1209	CNX	2
1210	M-IX	0
1212	MGMix Montgomery	5
1214	DE-CIX Dusseldorf	15
1228	CDIX	0
1229	CODIX	1
1235	MegaIX Perth	7