

Trabalho Prático 3

Sistema peer-to-peer de armazenamento chave-valor

Redes de Computadores - 2019/01
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais

Deiziane Natani da Silva
2015121980

1. Introdução

O trabalho consiste na implementação da funcionalidade básica de sistema chave-valor do tipo peer-to-peer, ou P2P. Através desse método, é possível encontrar o valor associado a uma chave específica, fazendo com que os programas que mantêm tal funcionalidade possam agir tanto como cliente quanto servidor. Tais sistemas são utilizados quando a prioridade é a descentralização de recursos e a sua heterogeneidade, visto em aplicações como BitTorrent e DNS.

2. Arquitetura

Nesse trabalho o protocolo utilizado foi o TCP, com conexões permanentes. Foi usada uma técnica de inundação semelhante à usada no OSPF. Toda vez que um *peer* receber uma mensagem de inundação ele decrementará seu TTL (*Time-to-Live*) e enviará para todas as outras peers que ela conhecer até que este TTL seja 0

3. Servent

O *servent* é o programa responsável pelo controle da troca de mensagens e o armazenamento da base de dados chave/valor, os quais estão em um arquivo de texto simples e foram implementados utilizando-se a estrutura de dados do tipo dicionário. Através do TCP é possível que haja conexões permanentes entre os *servents*.

Quando se é necessário propagar uma mensagem pela rede, o *servent* precisa primeiramente verificar se a mensagem nunca foi recebida por ele, com o objetivo de evitar duplicatas, utilizando uma lista de mensagens já recebidas pelo *servent* para esse fim. Caso ele já tenha recebido a mensagem, nada é feito. Porém, caso a

mensagem seja nova, o servent verifica se o TTL da mensagem, após decrementá-lo em uma unidade, é maior que 0. Caso isso se cumpra, ele armazena a mensagem como recebida e retransmite a mensagem para todos os servents vizinhos, além de, em alguns casos, enviar uma resposta ao cliente que enviou a requisição original. Sua implementação é feita utilizando-se duas classes: Servent e Message. A primeira é encarregada de representar cada um dos servents e possui métodos responsáveis pela estruturação da mensagem, como a construção da lista que guarda as mensagens recebidas e a inicialização do IP. Já a segunda representa a mensagem, dispondo métodos que têm a função de repassar a mensagem para os outros servents, decrementar o TTL e assim por diante

4. Client

O programa *Client* funciona como uma interface de rede para o usuário, permitindo que uma consulta por uma chave possa ser feita, além de ser possível saber como é a topologia de rede. A implementação é realizada a partir de duas classes: Client e Message. Client se encarrega de criar a estrutura do cliente, inicializando seu IP e seu PORT, além de setar o número de sequência para zero. A classe Message possui métodos que fazem a função de mandar as mensagens do tipo keyReq e topoReq, possuindo verificações de temporização para uma resposta à mensagem enviada chegar e imprimindo-a, após o tratamento e verificação dela.

O cliente faz a interação com o usuário a partir de uma interface com as seguintes opções:

- ? + chave = Consulta por uma chave
- T = Consulta a topologia da rede
- Q = Terminar

5. Discussão

O desenvolvimento do trabalho seguiu as orientações dadas pela especificação. O programa foi testado com várias entradas diferentes e argumentos na linha de comando. Todos os testes foram rodados com IP local (127.0.0.1) e portas diferentes.

6. Conclusão

O trabalho foi desenvolvido de maneira tranquila e serviu como um bom aprendizado a respeito da maneira como o protocolo TCP funciona e permitindo alcançar um grau de conhecimento maior com relação a aplicações tão usadas na Internet.