

Relatório

Trabalho 4 INF2545 – Sistemas Distribuídos

Aluno André Mazal Krauss, PUC-Rio, 2019.1

Professora Noemi Rodriguez

Propõe-se no trabalho aqui apresentado a implementação do algoritmo de *roteamento por boatos*¹ (Rumor Routing), usando o ambiente de programação Terra² em uma máquina virtual Ubuntu.

Foi implementada duas versões similares do programa, uma em que cada agente ou query está pré-programada, e uma em que é possível disparar novos eventos e queries interativamente, através da manipulação das condições de temperatura e luminosidade no simulador. As versões estão implementadas, respectivamente, nos arquivos `trab4_hardcoded.terra` e `trab4_interactive.terra`. Para executar os scripts é necessário usar o simulador e controlador Terra, disponíveis em [2]. Na versão interativa, para disparar um evento de temperatura, deve-se setar uma temperatura acima de 600 ou abaixo de 400, disparando um evento de temperatura alta ou baixa demais. Para disparar queries, deve-se similarmente manipular a leitura de luminosidade nos nós.

Implementação

Para se implementar o algoritmo proposto, primeiramente fez-se necessária a construção de uma lista de vizinhos, para cada nó. Isso é feito no início da execução do programa, em que cada nó manda um broadcast com o seu número identificador, e, simultaneamente, escuta por seus vizinhos, registrando cada um detectado na sua tabela. Após 10 segundos transcorridos, o programa segue com sua execução.

Para a funcionalidade em si do roteamento por boatos e detecção de eventos, utilizam-se 5 *tracks* em loop, executadas paralelamente. São elas:

1. **Receber um AGENTE**: *track* responsável por receber agentes de nós vizinhos, registrar o recebimento e repassá-los a outros vizinhos. É checado se o evento informado pelo agente já está registrado localmente e se ele apresenta um caminho mais curto ao evento. Somente é atualizado a tabela se o caminho do novo agente for de fato melhor. Além disso, se o número máximo de pulos para um agente não houver sido excedido, o agente é repassado a um vizinho aleatório diferente do vizinho de quem o nó recebeu o novo agente. Quando é registrado um evento em algum nó, é acendido um LED verde ou vermelho, para eventos de temperatura alta ou baixa.
2. **Receber uma QUERY**: *track* responsável por receber e tratar queries de nós vizinhos. Se o nó corrente não tiver qualquer informação sobre a query solicitada, ele simplesmente repassa a query a outro nó aleatório, diferente de quem o enviou a query. Ao mesmo tempo, ele registra o vizinho de quem veio a query, para construir um caminho de volta ao solicitante. O limite é de 100 pulos deste tipo. Se o nó corrente houver registrado um evento do tipo buscado, ele encaminha a query para o vizinho associado a esse evento. Se o nó corrente for a origem do evento buscado, ele adquire sua temperatura corrente e envia um *reply* ao vizinho que a solicitou.
3. **Receber um REPLY**: *track* responsável por repassar e tratar replies. Caso o nó corrente não seja o solicitante original, ele repassa o reply, pelo caminho salvo anteriormente.

Caso ele seja o solicitante, simplesmente registra a query como atendida, desligando o LED amarelo.

4. **Checar condições**: *track* responsável por checar os valores de temperatura e luminosidade, usados respectivamente para disparar eventos e queries. Essa *track* tem um cooldown de 5 segundos. Sem este cooldown, o simulador costumava travar, suponho que pelos contínuos cheques que eram feitos para cada nó na malha; além disso, em termos de sistema, de fato mais sentido checar periodicamente, e não continuamente.

As maiores dificuldades para a implementação foram entender as simplificações que deveriam ser feitas ao algoritmo para mante-se no escopo do trabalho e da linguagem Céu-T, por exemplo, considerar todos os eventos de um tipo como sendo o mesmo evento e não garantido o funcionamento se forem feitas diversas queries simultâneas. Para além disso, lidar com as limitações da linguagem e com o processo lento processo de execução dificultavam também.

Referências:

[1] – Paper em que é apresentado o algoritmo de *rumor routing*.

<https://dl.acm.org/citation.cfm?id=570742>.

[2] – Site de referência do ambiente Terra: <http://www.inf.puc-rio.br/~prjterra/>