Universidade Federal do Piauí
Centro de Ciências da Natureza
Departamento de Computação
Sistemas Distribuídos
Weslley Emmanuel Martins Lima
João Pedro dos Santos Patrocinio
30 de Novembro de 2020

Relatório da atividade de Multicast

Introdução

Esse relatório trata do desenvolvimento da atividade composta por uma aplicação que utiliza o protocolo Multicast para o envio e recebimento de mensagens. A aplicação envolve em construir um sistema tolerante a falhas utilizando múltiplos servidores para a realização de cálculos de expressões.

Instruções de Instalação

1. Para executar o programa, basta você executar o cliente / servidor usando o interpretador do **Python 3.**
2. O programa foi desenvolvido utilizando o **Python 3.8.5**, no Ubuntu 20.04 LTS. Todas as bibliotecas necessárias já estão contidas na instalação padrão do Python 3.
3. Você deve garantir que o tráfego de Multicast no grupo **224.14.0.244** e nas portas **1901** e **1902** esteja habilitado.

Instruções de Uso

1. Após instalar as dependências necessárias, chame o interpretador do Python 3 para inicializar a aplicação de servidor **server.py** em cada uma das máquinas voltadas para fazer o cálculo.
2. Agora, no cliente, abra a aplicação **client.py** e insira uma expressão numérica a ser calculada.
3. Um dos servidores terão de retornar o valor da expressão calculada e tal valor será mostrado na tela da aplicação cliente.

Implementação

Tanto a aplicação de cliente e de servidor utiliza as bibliotecas padrão do Python 3 para realizar o tráfego de mensagens UDP/Multicast na rede.

Na aplicação cliente, ele simplesmente envia uma mensagem multicast e espera por um retorno da expressão calculada, como se fosse um terminal que enviasse strings.

A aplicação de servidor, ao inicializar, procura definir o seu próprio ID na rede. Esse ID é definido intervalo de [1, ∞], obtido através do maior ID que responde na rede somado mais um (MaxID +1).
Após abrir o tal ID, ele cria uma thread responsável para responder às solicitações de Ping, enquanto que a thread principal se ocupa em responder às solicitações de cálculo.

Conclusão

Concluindo, foi possível criar um sistema de cálculo de expressões enviadas de um cliente para um dos múltiplos servidores responder. O servidor que deve responder é o de menor ID.

A tolerância a falhas existe por conta dessa redundância. A cada mensagem enviada pelo cliente, os servidores confirmam um com o outro para saber quem possui o menor ID, e daí responde-se essa mensagem.

Ainda pode se melhorar, por questões de conflitos de atribuição desses IDs, que pode ocorrer quando há apenas uma máquina e ela não responde aos multicasts de ping, fazendo que a nova máquina assuma o ID repetido e gere conflitos.