

Flower 1, 2, 3 Project: Compressed Network Communication

Projeto Final de Sistemas Operacionais

Débora Bianca Taveira de Moura
Ciência da Computação
Universidade Federal de Roraima
Boa Vista, Roraima - BR
Email: deborabiancatm@gmail.com

Ewelly Fabiane Cunha de Sousa
Ciência da Computação
Universidade Federal de Roraima
Boa Vista, Roraima - BR
Email: ewelly.fab@gmail.com

Abstract—Communication is the main purpose of technology development, and the necessity of reduce the quantity of data is an consequence of it. Currently we use differents applications to connect with anyone and anything, this consumes lot of space in databases or in cloud to save our data. Ocasionalmente the idea of compress data has created. On this project our purpose is create a compressed network communication between a server and a client using a TCP socket.

1. Introduction

O método utilizado para realizar a comunicação em nosso projeto é modelo cliente e servidor, onde o cliente é um processo que se conecta ao servidor para fazer uma requisição ou passar informações. Usualmente são criados agentes tanto do lado do cliente quanto do servidor para manipular a comunicação em rede com o intuito de proteger o aplicativo das complexidades dos protocolos de acesso remoto. Os agentes intermediários podem implementar recursos para aumentar a segurança, como criptografar o tráfego, e de redução do custo da comunicação, ao compactar o tráfego para melhorar o desempenho, de forma totalmente transparente para o usuário e o aplicativo, em nosso projeto realizaremos a compressão.

2. Metodologia

A network será composta pelos processos cliente, servidor, que são conectados via conexão TCP, e o shell que é conectado ao servidor via pipes. A figura 1 representa o esquema definido anteriormente. O servidor e o cliente serão desenvolvidos em C, a compressão será realizada com o auxílio da biblioteca Zlib, e as outras funcionalidades utilizaram as próprias bibliotecas do C.

2.1. Servidor

O programa do servidor executa em um soquete de rede com a porta especificada com o parâmetro port, ao se conectar com o cliente, recebe os comandos dele, atende os comando e os envia para o shell. Ele sempre aceitará uma

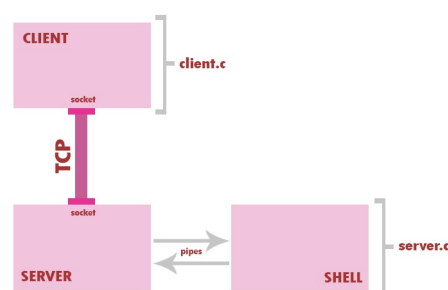


Figure 1. BigPicture

conexão e quando ela for feita é criado um fork do processo filho que executará um shell para processar os comandos recebidos.

O stdin/stdout/stderr é redirecionado do processo shell para as extremidades apropriadas do pipe.

2.2. Cliente

No programa cliente será aberta uma conexão com um servidor. O cliente ecoa a entrada do teclado para o soquete, e a entrada do soquete para o monitor; Possui também uma opção `-log=filename`, que mantém um registro dos dados enviados pelo soquete, salvando o histórico em um `.txt`, além da opção `-compress` ao cliente, para realizar a compressão dos dados;

O programa cliente abrirá uma conexão com um servidor passando a porta como parâmetro em vez de enviá-lo diretamente para um shell. O cliente deve então enviar a entrada do teclado para o soquete (enquanto ecoa para o monitor), e a entrada do soquete para o monitor. Dependendo dos parâmetros passados na conexão do do cliente ao servidor ele poderá executar algumas funções, como por exemplo:

- `-l`, `-log`, que gera um registro `.txt` dos dados enviados pelo soquete;
- `-c`, `-compress`, para realizar a compactação dos dados;
- `-p`, `-port` abre uma conexão com o servidor na porta especificada;

