Programação Paralela e Distribuída

Prof. Cidcley T. de Souza

Conteúdo

- Parte I
  - Introdução aos Sockets
  - Tipos de Sockets
  - Classes Java: InetAddress, Datagram Sockets, Stream Sockets, Output e Input Classes

Conteúdo

- Parte II
  - Criando Aplicações com Datagram Sockets
    - Exercício: BlackBoard
- Parte III
  - Criando Aplicação com Stream Sockets
    - Exercício: Talk

Sockets

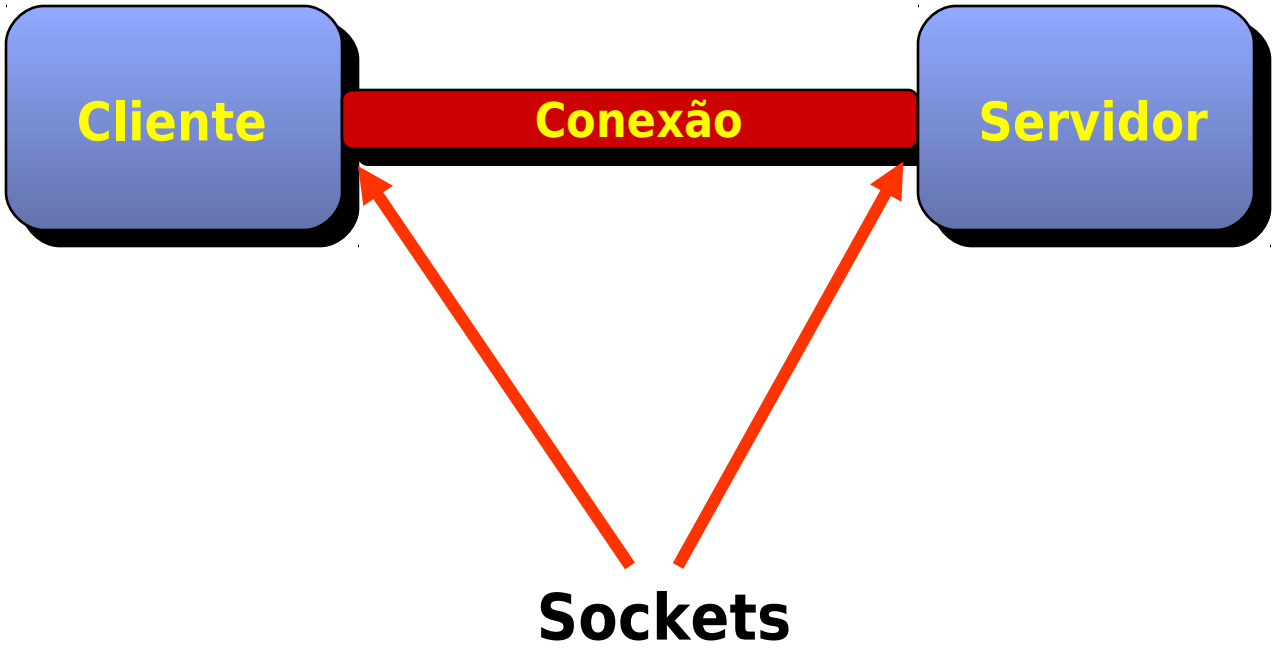
- Primeiro protocolo ponto-a-ponto para a comunicação implementado sobre TCP/IP;
- Introduzido em 1981 como interface genérica para IPC entre sistemas UNIX;
- Atualmente são suportados por virtualmente todos os sistemas operacionais;

O que são Sockets ?

- Sockets são pontos-finais (endpoints) para comunicação ponto-a-ponto entre sistemas;
- Esconde do programador alguns detalhes da rede;
- Implementado em três formas:
  - Stream: Interface para TCP
  - Datagram: Interface da UDP
  - Raw: Interface para ICMP

Sockets

O que são Sockets ?



Sockets

- Datagram
  - Interface para UDP (*User Datagram Protocol*);
  - Pacotes independentes sem garantia de entrega;

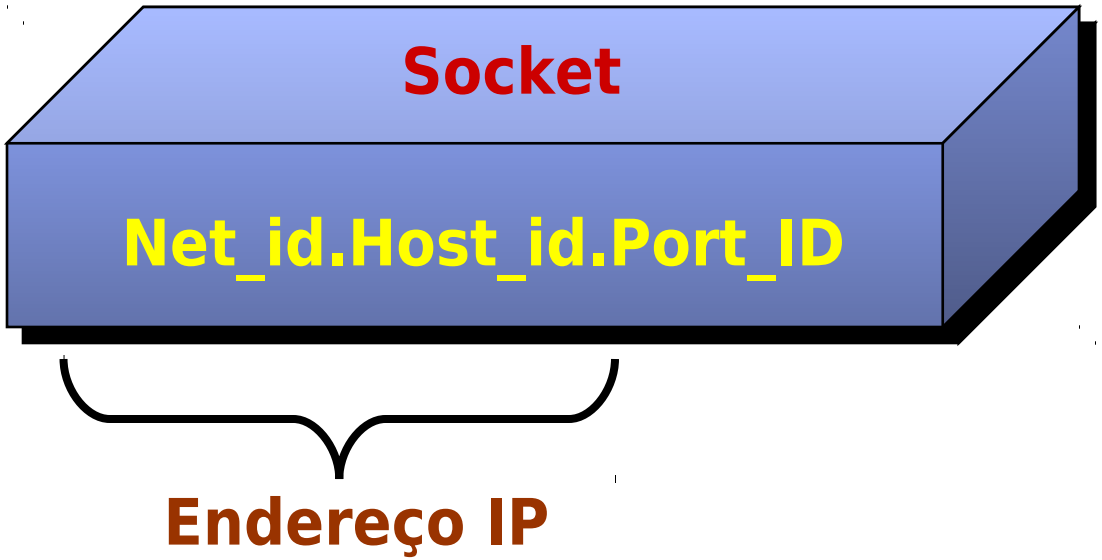
Sockets

- Stream
  - Interface para TCP (*Transmission Control Protocol*);
  - Transporte confiável, controle de fluxo, controle de sequenciamento de pacotes;

Sockets - Portas

- Cada Socket consiste em duas partes
  - Um endereço IP;
  - Um endereço de Porta;
  - Uma porta é uma entrada para uma determinada aplicação;
  - Existem portas reservadas (*well-known ports*) para serviços específicos;
    - Ex.:Porta 80 (Servidor HTTP)

Sockets - Portas



Socket = Endereço IP + Número de Porta

Sockets

- Uma associação é definida como uma quíntupla { protocolo, porta local, endereço local, porta remota, endereço remoto };

- Um conjunto de chamadas realizam a comunicação entre dois Sockets;

Java Sockets

- A Classe InetAddress
  - Classe utilizada para encapsular endereços IP;
  - Transforma um endereço IP em um objeto;
  - Permite se obter informações úteis sobre a rede invocando métodos desses tipos de objetos;

Java Sockets

- A Classe InetAddress
  - equals - Verifica se dois objetos representam um mesmo IP;
  - getAddress - Obtém o endereço IP de um objeto;
  - getByName - Determina o endereço IP de um host dado o seu nome;
  - getLocalHost - Obtém o IP do host local;

Java Sockets

- A Classe InetAddress (Exemplos)

```
//Retorna IP do Host “www.ifce.edu.br.br”
InetAddress ifce = InetAddress.getByName("www.ifce.edu.br.br");

//Retorna IP do End. “200.19.177.4”
InetAddress end1 = InetAddress.getByName("200.19.177.4");

//Retorna o IP do host local
InetAddress end2 = InetAddress.getLocalHost();
```

Java Sockets

- Classes para Datagramas Sockets
  - DatagramPacket
    - Implementa um objeto do tipo “pacote”;
    - Pacotes são enviados e recebidos pela rede utilizando sockets;
    - O construtor da classe é invocado para criar um pacote de saída, devendo-se ser especificado um array de bytes para ser enviado, bem como o endereço e a porta destino;

Java Sockets

- Classes para Datagramas Sockets
  - DatagramPacket
    - Pode também ser criado um pacote de entrada (recebimento) se utilizando um construtor e se especificando um array de bytes onde a informação recebida será armazenada;
    - O conteúdo do pacote pode ser retirado pelos métodos getData e getLength;

Java Sockets

- DatagramPacket – Construtores
  - Pacotes de Entrada
    - DatagramPacket (byte[] buffer, int length)
    - Dados do pacote são recebidos e armazenados em buffer (iniciando na posição buffer[0] e continuando até os dados serem totalmente armazenados ou length bytes terem sido escritos)

Java Sockets

- DatagramPacket – Construtores
  - Pacotes de Saída
    - DatagramPacket (byte[] data, int length, InetAddress destination, int port)
    - Pacote é preenchido com length bytes do vetor data (iniciando na posição data[0]); destination e port indicam o endereço de rede e a porta do destino para onde o pacote será enviado. length deve ser menor ou igual a data.length

Java Sockets

- DatagramPacket – Métodos de Instâcias
- InetAddress getAddress()
  - Devolve o endereço de rede (IP) definido para o pacote
- int getPort()
  - Devolve a porta associada ao pacote

Java Sockets

- DatagramPacket – Métodos de Instâcias
- byte[] getData()
  - Devolve um vetor de bytes contendo os dados do pacote
- int getLength()
  - Devolve o número de bytes dos dados do pacote (pode ser menor que getData().length)

Java Sockets (Criando um Pacote de Saída)

```
import java.net.*;
public class Pkt{
public static void main(String args[]){
    try {    // Define os dados do pacote a partir de uma string
             String s = "Ola Mundo!";
             byte[] dados = s.getBytes();

             // Define o IP e a porta a serem usados como destino do pacote
             InetAddress ip = InetAddress.getByName(“ifce.edu.br");
             int porta = 7;

             // Cria o pacote passando os dados da string, o IP e a porta
             DatagramPacket pacote = new DatagramPacket (dados, dados.length, ip,
    porta);

             // Mostra os dados do pacote na tela
             System.out.println(new String(pacote.getData(), "ASCII"));
         }catch (Exception e) { }
    }
}
```

Java Sockets

- Classes para Datagramas Sockets
  - DatagramSocket
    - Implementa um socket para envio e recebimento de datagramas;
    - Um socket pode ser criado passando-se o número da porta para o construtor;
    - Se nenhum número for especificado uma porta livre será utilizada;
    - O método getLocalPort retorna o número da porta alocada;

Java Sockets

- Classes para Datagramas Sockets
  - DatagramSocket
    - O método send deve ser invocado para se enviar um objeto do tipo DatagramPacket pelo socket;
    - O método receive deve ser invocado para se esperar por um pacote que deverá ser armazenado em um objeto do tipo DatagramPacket especificado;

Java Sockets

- Datagramas Sockets – Construtores
- DatagramSocket() throws SocketException
  - Cria um socket acoplado a um porta local qualquer (usado no lado do cliente)
- DatagramSocket(int port) throws SocketException
  - Cria um socket acoplado à porta especificada em port (usado no lado do servidor)

Java Sockets

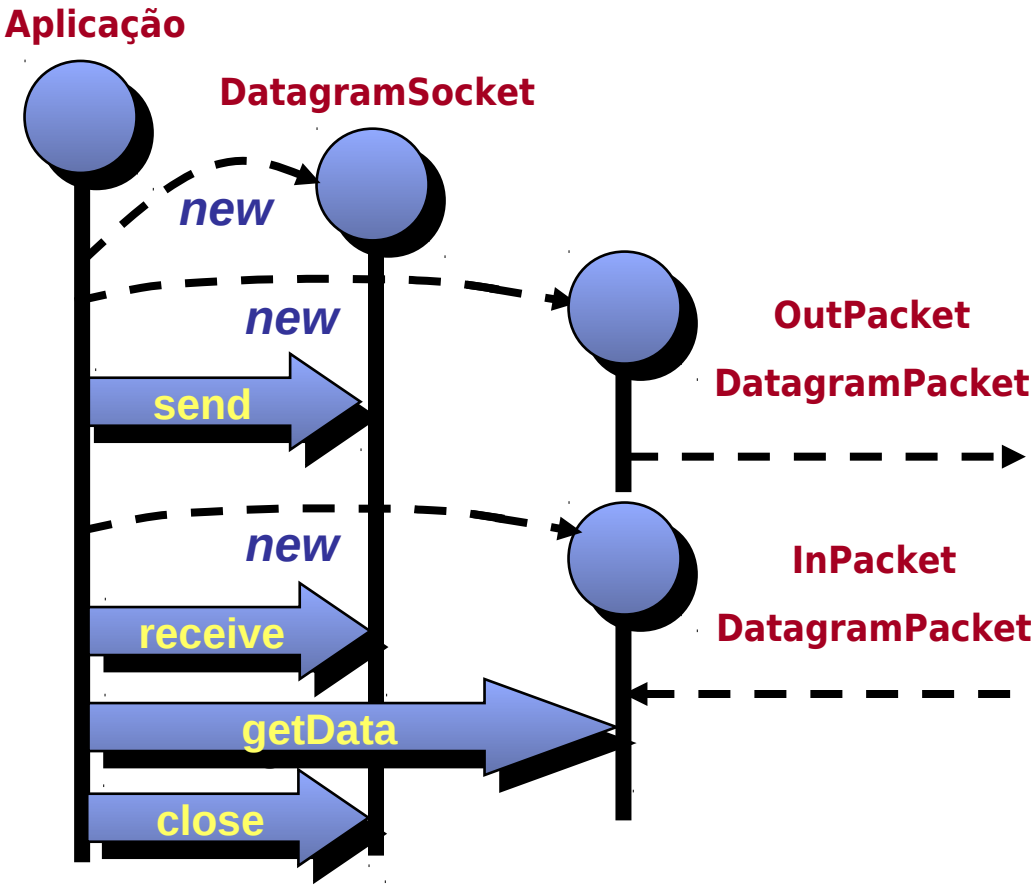
- Datagramas Sockets – Métodos de Instância
- void send(DatagramPacket dp)
  - Envia o pacote UDP especificado em dp para o seu endereço de destino (processo é liberado imediatamente após o pacote ser enviado)
- void receive(DatagramPacket dp)
  - Recebe um único pacote UDP da rede e o armazena no objeto especificado em dp (processo fica bloqueado até a chegada do pacote)

Java Sockets

- Datagramas Sockets – Métodos de Instância
- void close()
  - Fecha o socket e libera a porta à qual ele estava acoplado
- int getLocalPort()
  - Devolve o valor da porta local acoplada ao socket

Java Sockets

- Cenário Datagramas



Java Sockets (Enviando Pacotes - 1/2)

```
import java.net.*;
import java.io.*;
import java.util.Scanner;
public class envia{
public static void main(String args[]){
try {
   // Define um endereço de destino (IP e porta)
   InetAddress servidor = InetAddress.getByName("localhost");
   int porta = 1024;
   // Cria o socket
   DatagramSocket socket = new DatagramSocket();
   // Laço para ler linhas do teclado e enviá-las ao endereço de
   destino
   Scanner input = new Scanner(System.in);
   String linha = input.nextLine();
```

Java Sockets (Enviando Pacotes - 2/2)

```
   while (!linha.equals(".")) {
         // Cria um pacote com os dados da linha
         byte[] dados = linha.getBytes();
         DatagramPacket pacote = new DatagramPacket(dados,
   dados.length, servidor, porta);
         // Envia o pacote ao endereço de destino
         socket.send(pacote);
         // Lê a próxima linha
         linha = input.nextLine();
   }
   } catch (Exception e) {}
}

}
```

Java Sockets (Recebendo Pacotes - 1/2)

```
import java.net.*;
import java.io.*;
public class recebe{
public static void main(String args[]){

    int porta = 1024; // Define porta
    byte[] buffer = new byte[1000]; // Cria um buffer local
    try {
    // Cria o socket
    DatagramSocket socket = new DatagramSocket(porta);
    // Cria um pacote para receber dados da rede no buffer local
    DatagramPacket pacote = new DatagramPacket(buffer,
    buffer.length);
```

Java Sockets (Recebendo Pacotes - 2/2)

```
// Laço para receber pacotes e mostrar seus conteúdos na saída padrão
    while (true) {
        socket.receive(pacote);
        String conteudo = new String(pacote.getData(), 0, pacote.getLength());
        System.out.println("End. Origem: " + pacote.getAddress());
        System.out.println("Conteudo Pacote: " + conteudo + "\n");
        // Redefine o tamanho do pacote
        pacote.setLength(buffer.length);
    }
  } catch (Exception e) {}
 }
}
```