Java RMI

Arthur Haas Dorneles

Java RMI

- RMI (Remote Method Invocation).
- Dividido em 3 camadas:
  - Stub/Skeleton
  - Referências Remotas
  - Transporte

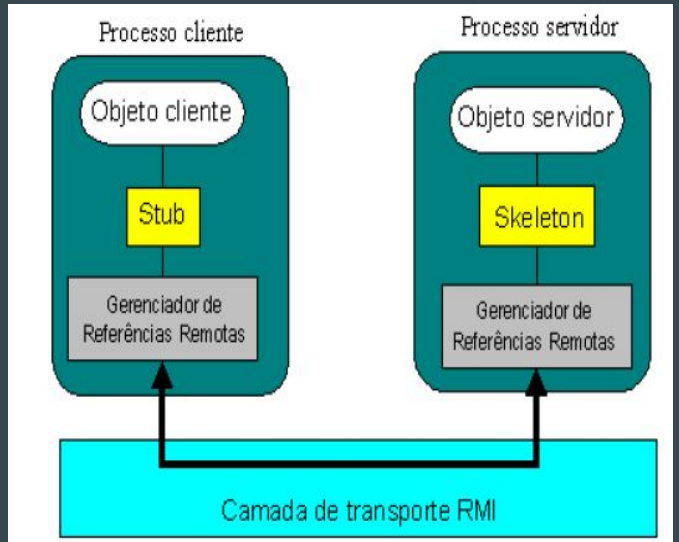
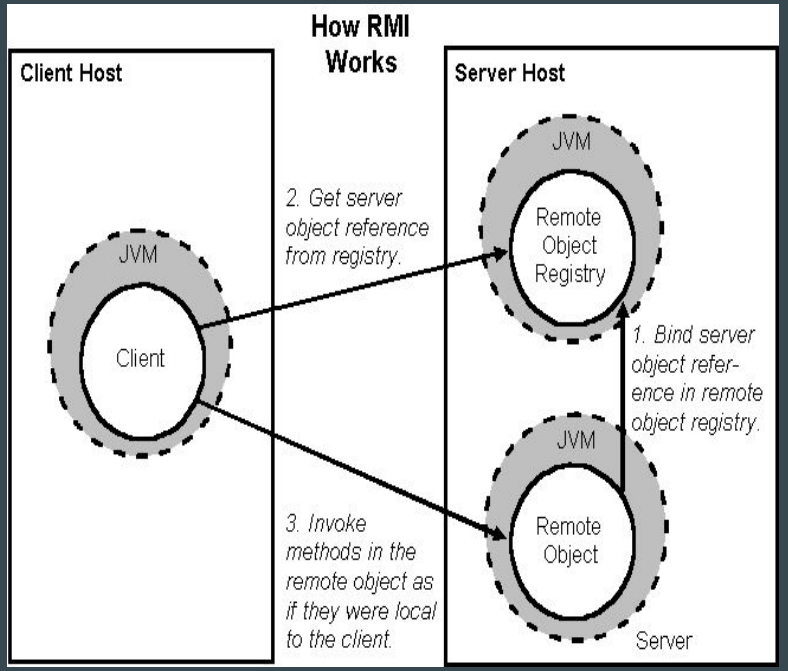


Ilustração 1: Arquitetura RMI

Fonte: http://www.dca.fee.unicamp.br/cursos/PooJava/objdist/rmiarq.html

Como funciona?

- Stub e Skeletons (Marshalling)
- RRL (Locate Registry)
- Transporte
  - Endpoint
  - Transporte
  - Canal
  - Conexão



Implementação

- Calculadora que pode realizar somas, subtrações, multiplicações e divisões
- Interface da Calculadora:

```java
import java.rmi.Remote;

public interface Calculadora extends Remote{
    public double calc(double a,double b, char c) throws RemoteException;
}
```

Implementação

- Implementação da interface Calculadora:

```java
import java.rmi.RemoteException;

public class CalculadoraImple extends UnicastRemoteObject implements Calculadora{

    private static final long serialVersionUID = 1L;

    protected CalculadoraImple() throws RemoteException{
        super();
    }
    public double calc(double a, double b, char op) throws RemoteException{
        double i = 0;
        if(op == '+') {
            i = this.add(a, b);
        }else if(op == '-') {
            i = this.sub(a, b);
        }else if(op == '*') {
            i = this.mult(a, b);
        }else if(op == '/') {
            i = this.div(a, b);
        }
        return i;
    }
    private double add(double a, double b){
        System.out.println("A = " + a + " B = " + b);
        return a+b;
    }
    private double sub(double a, double b){
        System.out.println("A = " + a + " B = " + b);
        return a-b;
    }
    private double mult(double a, double b){
        System.out.println("A = " + a + " B = " + b);
        return a*b;
    }
    private double div(double a, double b){
        System.out.println("A = " + a + " B = " + b);
        return a/b;
    }
}
```

Implementação

- Servidor:

```java
import java.rmi.Naming;

public class CalculadoraServer {
    CalculadoraServer() {
        try {
            System.setProperty("java.rmi.server.hostname","192.168.0.15");
            LocateRegistry.createRegistry(1099);
            Calculadora c = new CalculadoraImple();
            Naming.bind("CalculadoraService", (Remote) c);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        new CalculadoraServer();
    }
}
```

Implementação

- Cliente:

```java
import java.rmi.Naming;
public class CalculadoraClient {
    public static void main(String[] args) {
        try {
            Calculadora c = (Calculadora) Naming.lookup("rmi://192.168.0.15:1099/CalculadoraService");
            Scanner l = new Scanner(System.in);
            double a,b;
            char op;
            int d = 0;
            while(d == 0){
                System.out.println("0-Calculadora\n1-Sair");
                d = l.nextInt();
                if(d == 0){
                System.out.println("Informe os valores a serem calculados e a operação na seguinte forma -> <Valor1> <Operação> <Valor2>:");
                System.out.println("Valor 1");
                a=l.nextDouble();
                System.out.println("Operador");
                op = (char)System.in.read();
                System.out.println("Valor 2");
                b=l.nextDouble();
                System.out.println("Resultado : " + c.calc(a,b,op));
                }
            }
            l.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Execução

- Compilar (javac)
- Gerar stub (rmic)
- Executar (java)