Java RMI

Marcos Antonio de Oliveira Dias
Escolas e Faculdades QI - Curso de Análise e Desenvolvimento de Sistemas

maantoniodias@gmail.com

O objetivo deste artigo é apresentar um conceito básico sobre o java RMI (Remote Method invocation), nele será abordado alguns objetivos dentro da programação onde o RMI absorve a execução de objetos em um único computador ou computadores distantes dentro de uma plataforma Java "comando métodos remotos."

1. INTRODUÇAO

Sistemas de chamada de procedimento remoto existem desde meados de 1984, quando foram propostos pela primeira vez (AD Birrell e B. J. Nelson, 1984). Durante os intervenientes 15 anos, numerosos melhoramentos evolutivos ocorreram no sistema de base de RPC, conduzindo a sistemas, tais como melhoradas NCS (T. H. Dineen et al., 1987).

Para que possamos executar objetos Java em um computador ou talvez em diversos computadores a RMI fará a comunicação deles através de apontamentos de códigos remotos. Uma tecnologia semelhante foi base para a RMI, assim denominada de chamada de procedimento remoto (RPC). Essa tecnologia admite a programação de funções partindo de outra maquina, imaginado que a função estivesse sendo executada nesta mesma. [Deitel 2004]. Podemos dizer que a RMI é a implantação da RPC por Java para comunicação distribuída de um objeto Java com outro. Acessando um método ou serviço Java esse será registrado remotamente, assim um fornecedor poderá fazer uma pesquisa e ter uma resposta de permissão ou não para sua solicitação de acesso de utilização.

2. CAMADAS RMI

As operações e soluções que utilizamos no processo de programação podem estar ocultas ou escondidas e o responsável por isto é o RMI. Dividindo o sistema RMI vamos verificar três camadas onde estão listadas e ordenadas devido à proximidade de quem está programando (figura 1):

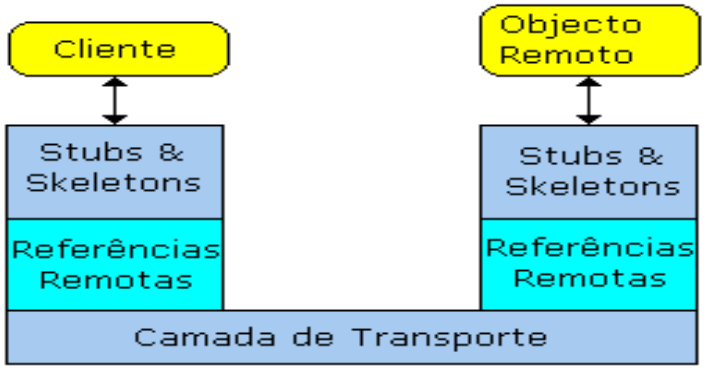

Figura 1. Camadas de Transporte

1. A camada de *stub*/*skeleton* é responsável por receber as chamadas da aplicação cliente, após realiza á interface e reencaminha para o objeto remoto;
2. A camada de Referências Remotas (Remote Reference Layer) lida com a gestão e com a interpretação das referências remotas;
3. A camada de Transporte assegura a ligação entre as máquinas virtuais através de TCP/IP. (Sarmento, 2003);

3. FUNCIONAMENTO

“O funcionamento de RMI consiste basicamente em dois programas, um seria o cliente e outro o servidor. O servidor instancia objetos remotos, o referencia com um nome e faz um "BIND" dele em uma porta, onde este objeto espera por clientes que invoquem seus métodos. Já o cliente referencia remotamente um ou mais métodos de um objeto remoto. RMI fornece os mecanismo para que a comunicação entre cliente e servidor seja possível” . (Domingues, 2000).

Aplicações distribuídas precisam, portanto, executar as seguintes ações:

- *Localizar Objetos Remotos* - Uma aplicação pode usar dois mecanismos para obter referências de objetos remotos. Ela pode registrar o objeto remoto com a ferramenta de nomes do RMI, que se chama "rmiregistry", ou ela pode passar e retornar referências aos objetos remotos como parte de sua operação normal.

- *Se Comunicar com Objetos Remotos* - Os detalhes de comunicação entre objetos remotos são tratados pelo RMI, ou seja, para o programador, a comunicação remota é semelhante a uma chamada ao método localmente.

- *Carregar "bytecodes" de objetos móveis* - Como o RMI permite que objetos remotos sejam passados como parâmetros numa função, ele fornece os mecanismos necessários para carregar o código dos objetos remotos.

- A figura Abaixo ilustra como uma aplicação RMI distribuída usa o "rmiregistry" para obter uma referência ao objeto remoto. O servidor chama o registro para associar um nome ao objeto remoto. O cliente procura o objeto remoto pelo seu nome no registro do servidor e então invoca um método nele. A ilustração também mostra que o RMI pode utilizar um servidor "web" para carregar os "bytecodes", de servidor para cliente e vice-versa, de acordo com as necessidades da aplicação (Domingues, 2000).

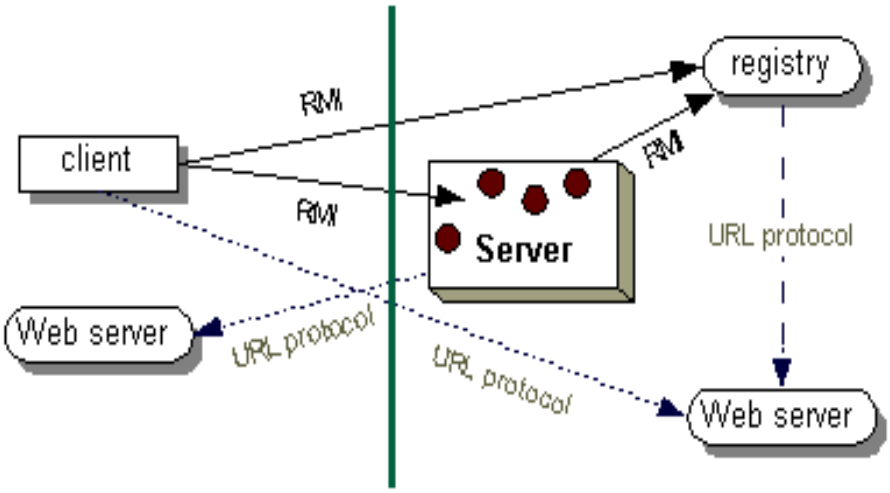

Figura 2: RMI Registry da Aplicação Distribuída

4. Vantagens de Carregar o Código Dinamicamente:

Uma das principais vantagens do RMI é sua capacidade de baixar o código de um objeto, caso a classe desse objeto não seja definida na máquina virtual do receptor. Os tipos e o comportamento de um objeto, previamente disponíveis apenas em uma máquina virtual, agora podem ser transmitidos para outra máquina virtual, possivelmente remota. Essa funcionalidade do RMI permite que o código da aplicação seja atualizado dinamicamente, sem a necessidade de recompilar o código.

5. CONCLUSÃO

Onde queremos uma comunicação eficiente entre processos remotos o RMI é a ferramenta indicada. Utilizando esta tecnologia podemos solucionar problemas ou dificuldades com um grau maior de certeza com relação ao uso de RPC, onde neste, não responde a dados mais complexos e assim direciona o desenvolvedor a se aprofundar em uma IDL especifica.

Podemos citar varias vantagens, mas as que mais se diferenciam são: poder de transparência de comunicação e o seu uso simplificado. Como esta utiliza a linguagem conceitual Java, o RMI pode criar aplicações com facilidade.

5. REFERÊNCAS BIBILIOGRÁFICAS

Domingues, Vicente Barreto. *SOA RMI (Remote Method Invocarion)*
www.gta.ufrj.br/grad/00_1/vicente/rpc.htm
Acesso em: 20 de Agosto de 2017

Sarmento, Luiz . *Breve Introdução ao RMI - Remote Method Invocarion*
https://web.fe.up.pt/~eol/AIAD/aulas/JINIdocs/rmi1.html
Acesso em: 25 de Setembro de 2017

Gomes, Davi . *Uma Introdução ao RMI em Java*
http://www.devmedia.com.br/uma-introducao-ao-rmi-em-java/28681
Acesso em: 20 de Outubro de 2017.

Castro Marcio, Raeder Mateus, Nunes Thiago. *RMI Uma Visão Conceitual*
inf.pucrs.br/~gustavo/disciplinas/sd/material/Artigo_RMI_Conceitual.pdf
Acesso em: 20 de Outubro de 2017