



Universidade Federal Do Rio Grande Do Norte

Escola Agrícola de Jundiaí

Disciplina: Sistemas Distribuídos

Docente: Taniro Chacon

**Implementação e análise de mecanismos de comunicação  
Sockets (UDP e TCP), RMI e MOM**

João Gabriel Quaresma

Fábio Henrique

Macaíba/RN

2018

**Indrodução**

A implementação consiste na solução de duas problemáticas empregando o modelo de sistema distribuído cliente-servidor, em cada um deles, utilizando como métodos, sockets (UDP e TCP), RMI e MOM (usando a *API Apache ActiveMQ™* desenvolvida pela *Apache Software Foundation*). Os problemas consistem em, o primeiro a resolução de uma conta simples de raiz quadrada e o segundo a medição da banda passante da rede, ambos problemas tem o intuito de se obter o tempo em milisegundos do envio e recebimento de dados e com a amostra de dados obtida calcular o desvio padrão para análise da dispersão dos dados. Abordaremos a seguir os seguintes casos de estudo: descrição do ambiente de testes, hipótese, teste de hipótese, metodologia e análise dos resultados.

Abordaremos a seguir os seguintes casos de estudo:

**Descrição do ambiente de testes**

O ambiente de testes foi realizado em um computador com as seguintes configurações:

* Processador: *Intel Core i5 7500 @ 3.40GHz*
* Memória RAM: *8GB*
* Placa de vídeo: *NVIDIA GeForce GTX 1060 6GB*
* Sistema Operacional: *Windows 10*

**Hipótese**

A hipótese que tivemos consiste em provar, com dados estatísticos que sockets de baixo nível de abstração, UDP e TCP possuem menor tempo de e envio e resposta em relação outros mecanismos de comunicação como *Remote Method Invocation (RMI)* e *Message Oriented Middleware (MOM).*

**Metodologia**

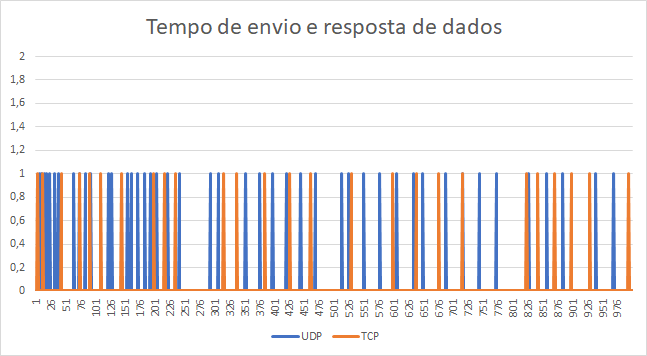
Como metodologia realizamos envios e recebimentos de dados, em um laço de repetição que é executado mil vezes, que nele, a cada volta é calculado o tempo do transporte dos dados e com isso obtivemos uma amostra de dados que nos possibilitou poder de traçar gráficos e calcular o desvio padrão de cada mecanismo de comunicação, já citados anteriormente.

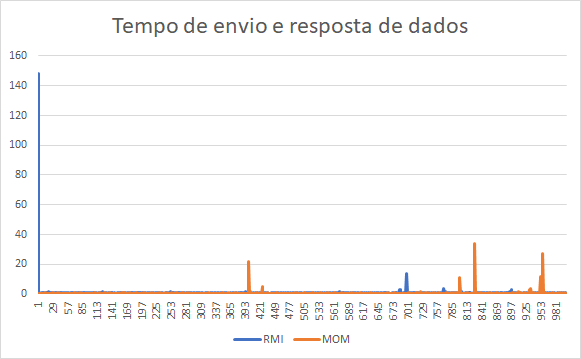
Na questão 1 foi utilizado a função *currentTimeMillis()*, que nada mais é o retorno de um tempo em milissegundos. Enquanto, na questão 2 foi utilizado a função *nanoTime()*, que nada mais é o retorno de um tempo em nanosegundos.

**Teste de Hipótese**

Tabela de testes da questão 1:

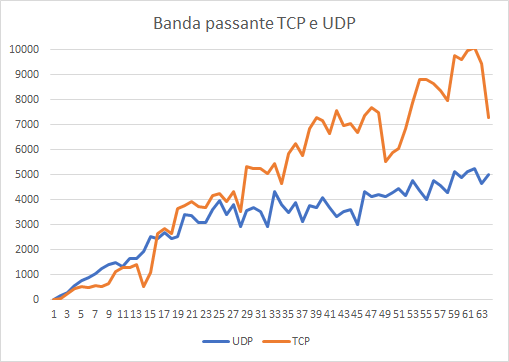
|  |  |  |
| --- | --- | --- |
|  | **Média** | **Desvio Padrão** |
| **UDP** | 0,046 | 0,20959 |
| **TCP** | 0,026 | 0,159215 |
| **RMI** | 0,607 | 4,716651 |
| **MOM** | 0,19 | 1,651856 |

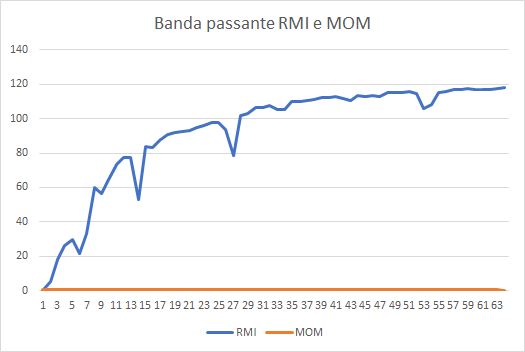




Testes da questao 2:

|  |  |  |
| --- | --- | --- |
| **Banda passante** | **Média** | **Desvio Padrão** |
| **UDP** | 3210 | 1360,724 |
| **TCP** | 4937,546 | 2992,949 |
| **RMI** | 92,63003 | 30,75824 |
| **MOM** | 0,557495 | 0,12513 |

****

****

**Análise dos resultados**

Observamos com os testes realizados e podemos que o melhor modelo de comunicação é o socket TCP devido ao seu tempo de envio e recebimento de dados ser menor que os demais, proveniente ao seu menor nível de abstração.

Então podemos concluir que no final das contas, quanto maior o nível de abstração para implementação e execução de um algoritmo de comunicação para sistemas distribuídos, maior será o tempo de execução e menor será o desempenho para a troca de dados/informações. E com isso posto, podemos concluir que uso de determinada tecnologia ou metodologia de implementação, vai de acordo com a problemática e sua respectiva solução, pois sempre terá que haver o estudo dos parâmetros a serem considerados - velocidade, desempenho e integridade dos dados que serão enviados e recebidos.