

# Relatório de Computação II

## Load Balancing

Bruno Daiki YAMADA

2 de março de 2016

Data do procedimento: 27 de fevereiro de 2016  
Professor: Yoshiharu Kohayakawa

## 1 Objetivos

Implementar e analisar duas rotinas diferentes de distribuição de usuários em servidores. Para isso foi implementada uma classe disponível no site *Server.java*, que utiliza queues para representar a lista de usuários utilizando cada server.

## 2 Procedimento

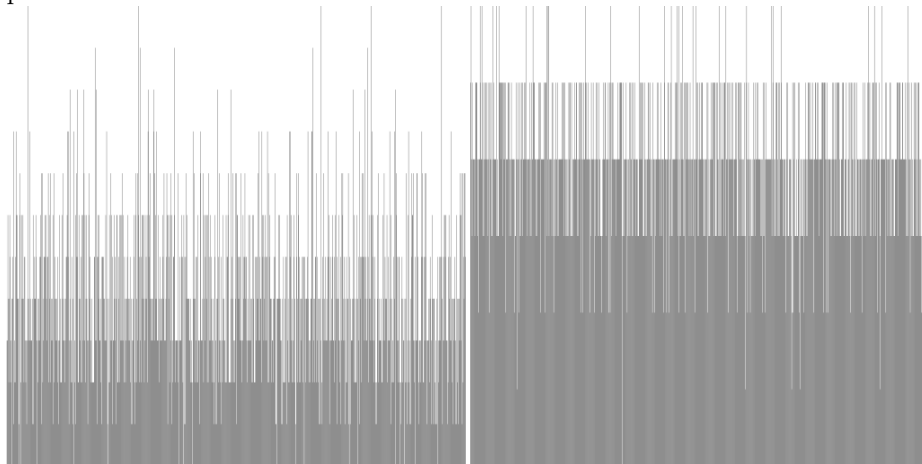
Após uma análise superficial, concluí que a classe *Server.java* carecia uma função que retornava a carga do server, então complementei o script com essa função. Então escrevi dois programas que implementam as rotinas descritas nos Web Exercises 43 e 44 em *ServerRoutine.java* e *ServerRoutine2.java* respectivamente. Executando as classes individualmente é possível de controlar diretamente a quantidade de servers e usuários na linha de comando, plotando o gráfico automaticamente. Um terceiro programa, *ServerBenchmark.java* foi escrito para analisar a *maxload* de cada rotina, dobrando o quantidade de servers e usuários a cada iteração.

### 3 Resultados e Discussão

```
C:\Users\cezar\Documents\Java2>java-algs4 -Xmx1g ServerBenchmark
Routine 1 : Routine 2
2 1 : 1
4 2 : 1
8 3 : 2
16 3 : 2
32 4 : 3
64 5 : 3
128 4 : 2
256 4 : 3
512 5 : 3
1024 5 : 3
2048 6 : 3
4096 6 : 3
8192 8 : 3
16384 7 : 3
32768 9 : 3
65536 7 : 3
131072 9 : 4
262144 8 : 4
524288 10 : 4
1048576 9 : 4
2097152 9 : 4
4194304 10 : 4
8388608 12 : 4
```

Reparamos na tabela acima que claramente temos uma diferença de *max load* entre as duas estratégias, especialmente para valores muito altos, como no caso mais de 8 milhões de usuários e servers, tendo uma carga máxima três vezes maior na rotina 1.

Os dois gráficos abaixo são um gráfico de barras representando a carga de cada um dos servers, e ilustra a diferença de distribuição entre as duas estratégias para  $N = 1000$  e  $M = 4000$ :



Entretanto, não foi possível verificar a hipótese fornecida pelo enunciado sobre *maxload* em função de  $N$ . Pois para  $N = 8388608$ , temos:

$$MaxLoad_1 = \log(8388608)/\log(\log(8388608)) \approx 5,75 \neq 12$$

$$MaxLoad_2 = \log(\log(8388608)) \approx 2,77 \neq 4$$

## 4 Conclusão

A tabela gerada pelo *ServerBenchmark.java* demonstra que existe uma clara diferença entre  $\log(\log(n))$  e  $\log(n)/\log(\log(n))$  levando uma grande vantagem de distribuição com uma modificação pequena no algoritmo, aumentando o tempo de tempo de execução somente para geração de números gráficos de carga demonstram que a demanda está muito melhor distribuída na rotina do exercício 44.