

Processamento Paralelo e Distribuído

Relatório

Trabalho 1

Ataque de Dicionário

Nome: Wanderson Ralph Silva Vita

Matrícula: 2013101887

Professor: João Paulo

Sumário

Processamento Paralelo e Distribuído	1
Relatório	1
Trabalho 1	1
Ataque de Dicionário	1
Sumário	2
Configurações	3
Executar	3
Compilar	3
Executar Metres	3
Executar Escravo	3
Executar Cliente	3
Para rodar a aplicação siga essa ordem	3
Tempo de Resposta	4
Speed Up	5
Eficiência	5
Overhead	6

Configurações

Os fontes possi o arquivo de configuração `config.properties` onde pode ser configurado o ip do mestre, além do nome do mesmo, tempos de checkpoint e monitoramento.

```
trab1 > properties > config.properties
1  master.name = mestre
2  run.parallel = true
3  size.vector = 1000
4  timer.monitor = 20
5  timer.checkpoint = 10
6  timer.addSlave = 30
7  server.hostname = localhost
```

Executar

O código fonte possui um arquivo `makefile` que permite compilar, e executar o mestre, escravo e cliente.

Compilar

make

Executar Metres

make master

Executar Escravo

make slave

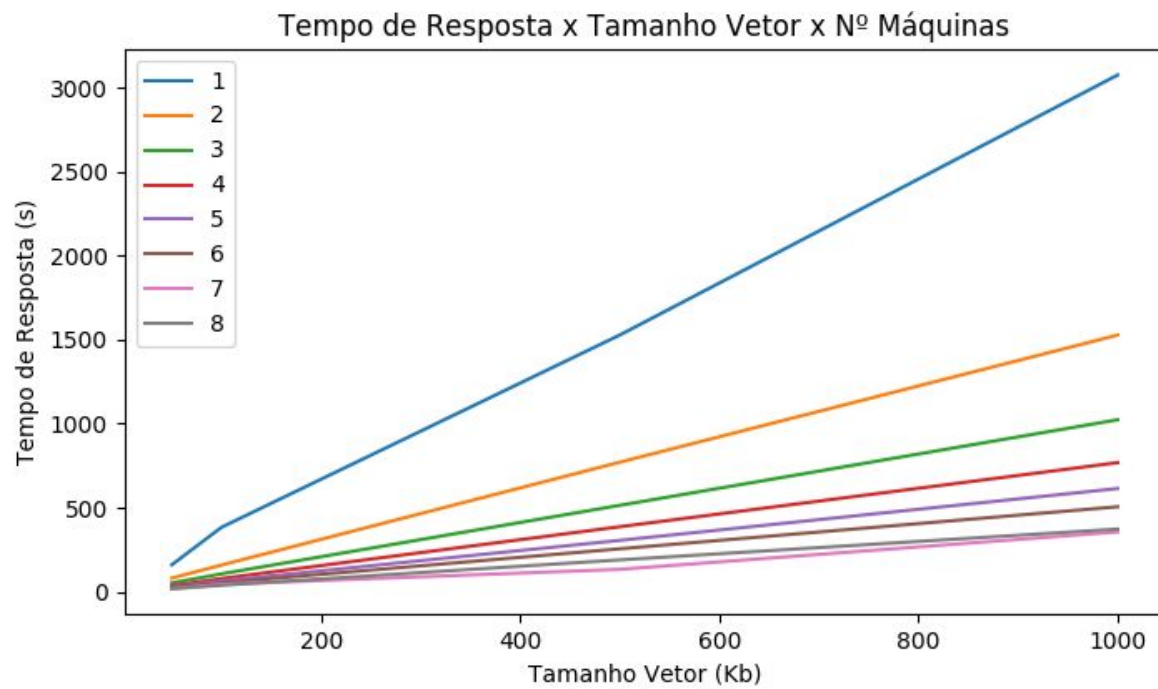
Executar Cliente

make client

Para rodar a aplicação siga essa ordem

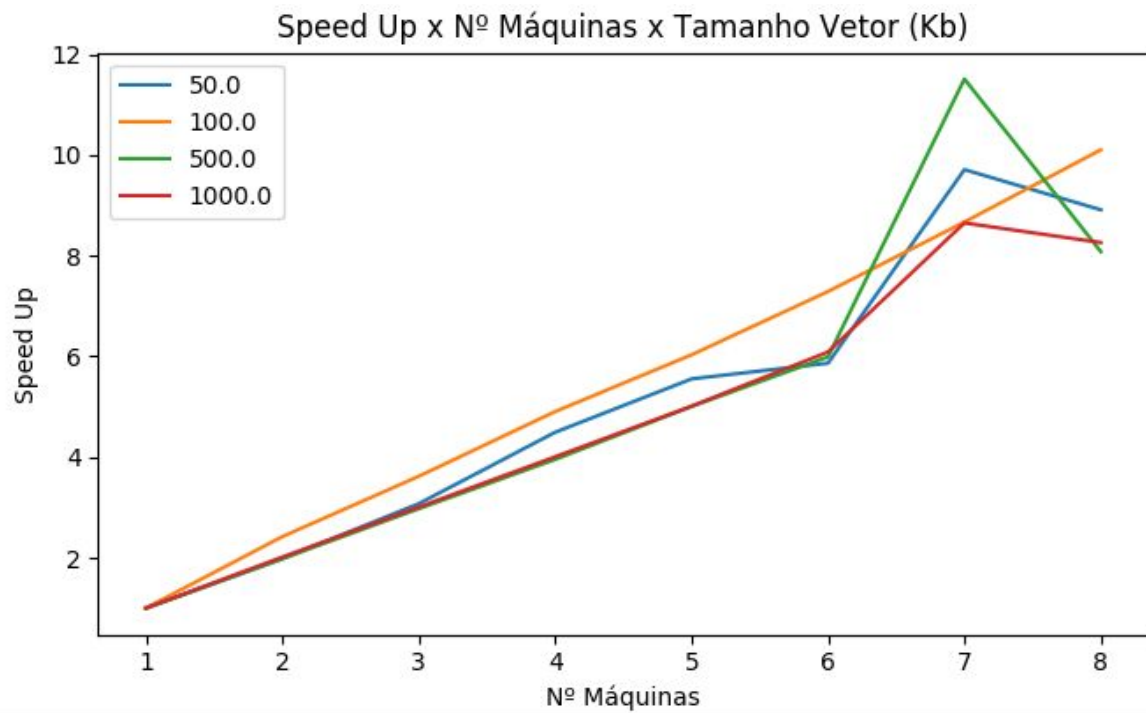
1. *make*
2. Terminal 1
 - a. *cd bin*
 - b. *rmiregistry*
3. Terminal 2
 - a. *make master*
4. Terminal 3 à n
 - a. *make slave*
5. Terminal n+1
 - a. *make client*
 - i. Esse já passa um arquivo e knotext de teste.

Tempo de Resposta

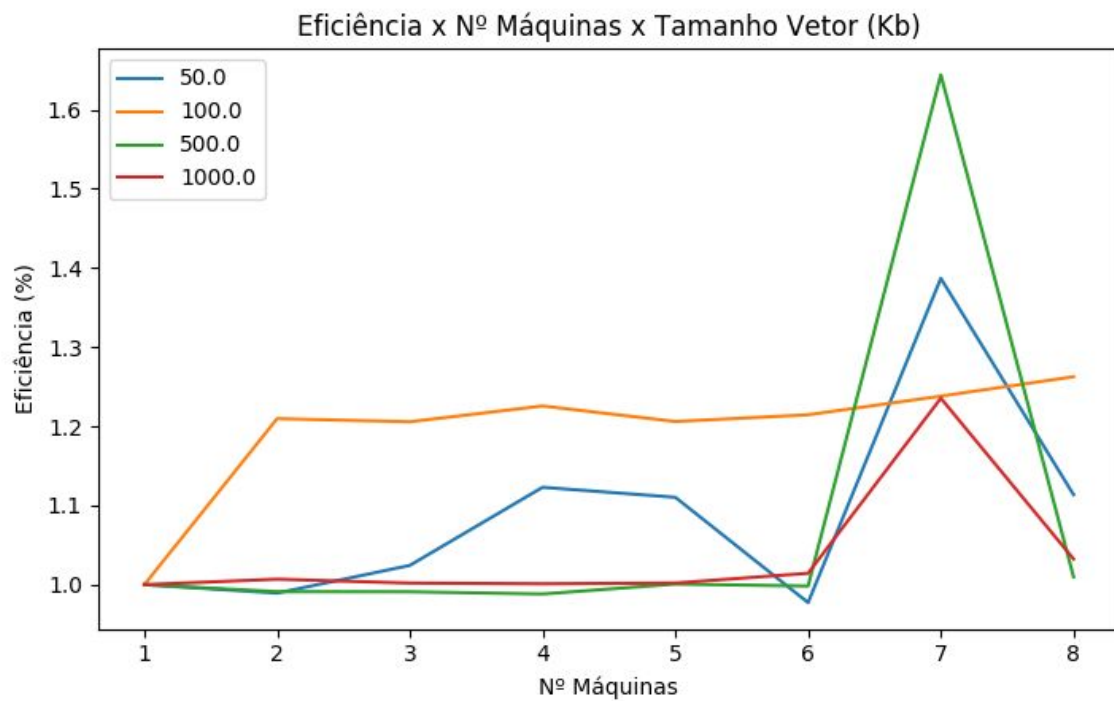


Como esperado, o tempo de resposta diminui conforme aumenta o número de máquinas. E conforme vai aumentando, a diferença de tem vai diminuindo.

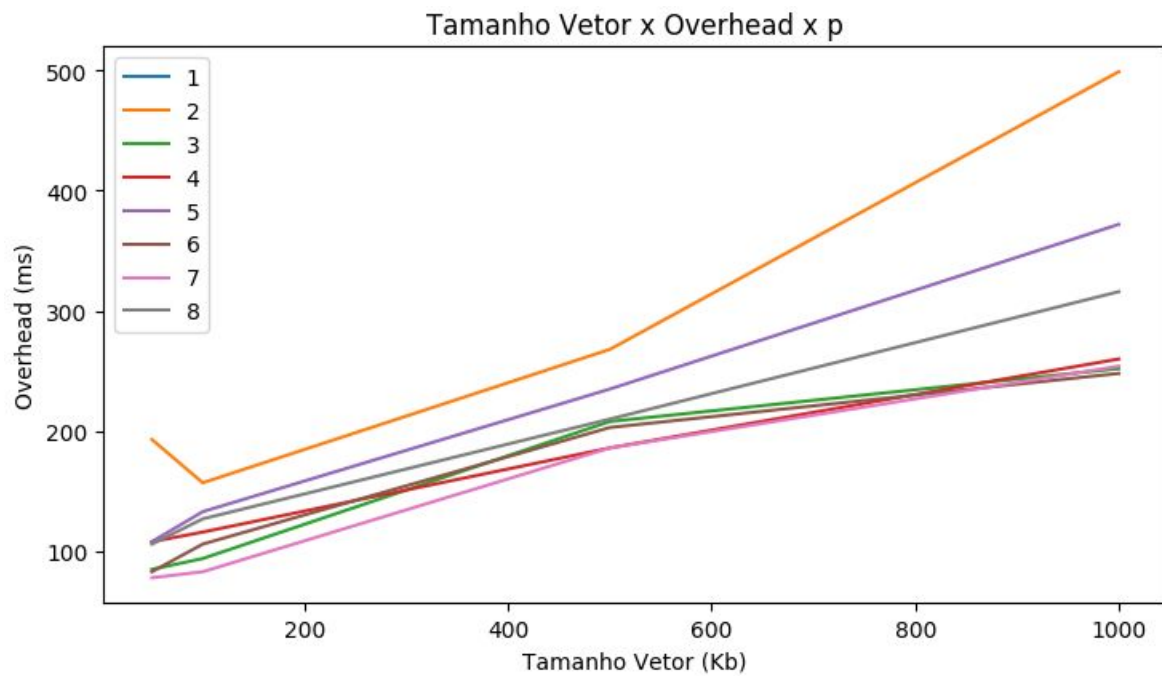
Speed Up



Eficiência



Overhead



Intuitivamente com o aumento do número máquinas, o overhead deveria ser maior. Mas como o vetor é dividido em faixas iguais de mil em mil, isso permite que enquanto espera resposta de uma, faça requisição das outras, fazendo com que o tempo de overhead de uma requisição seja preenchido com processamento de outras requisições.