MAC0438 - Programação Concorrente
Exercício Programa 1
Multiplicação de Matrizes

André Yoshiaki Kashiwabara [*]
Carlos Augusto Leite [†]
Professor Alfredo Goldman

8 de abril de 2003

[*] NUSP: 3370552, <yoshiaki@linux.ime.usp.br>
[†] NUSP: 3157672, <leite@linux.ime.usp.br>

Sumário

1 Introdução

O exercício programa foi testado tanto em máquinas monoprocessadas quanto em máquinas multiprocessadas, os resultado dos teste serão mostrados a seguir por meio de gráficos, sendo especificada para cada máquina sua configuração, nome e local bem como as obervações sobre os gráficos obtidos.

2 Máquinas monoprocessadas

As máquinas monoprocessadas poderão ser dividas de acordo com o sistema operacional sobre o qual a máquina está executando o exercício programa.

2.1 Máquina Curupira - Windows

2.1.1 Local

A máquina Curupira está localizada no bloco do Instituto de Matemática e Estátisca da Universidade de São Paulo à sala 259-B, onde se encontra o Laboratório de Banco de Dados - BIOINFO.

2.1.2 Configuração

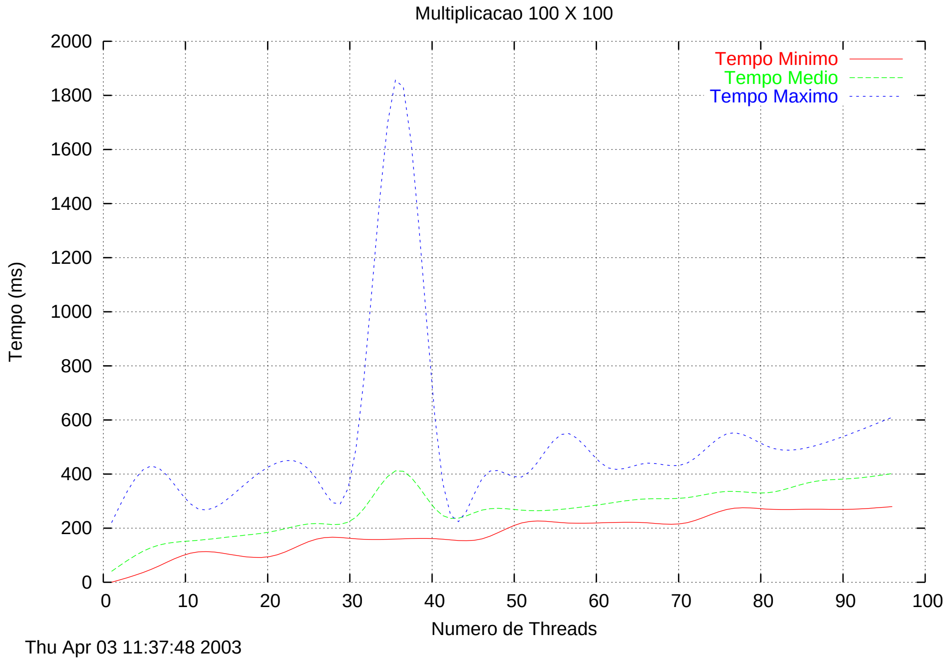
A máquina Curupira possui um processador AMD de 1.8 Ghz, 654 Mb de memória RAM, 40 Gbytes de hard disk e funciona sobre o sistema operacional Windows 98 (o que é um desperdicio - em nossa opinião).

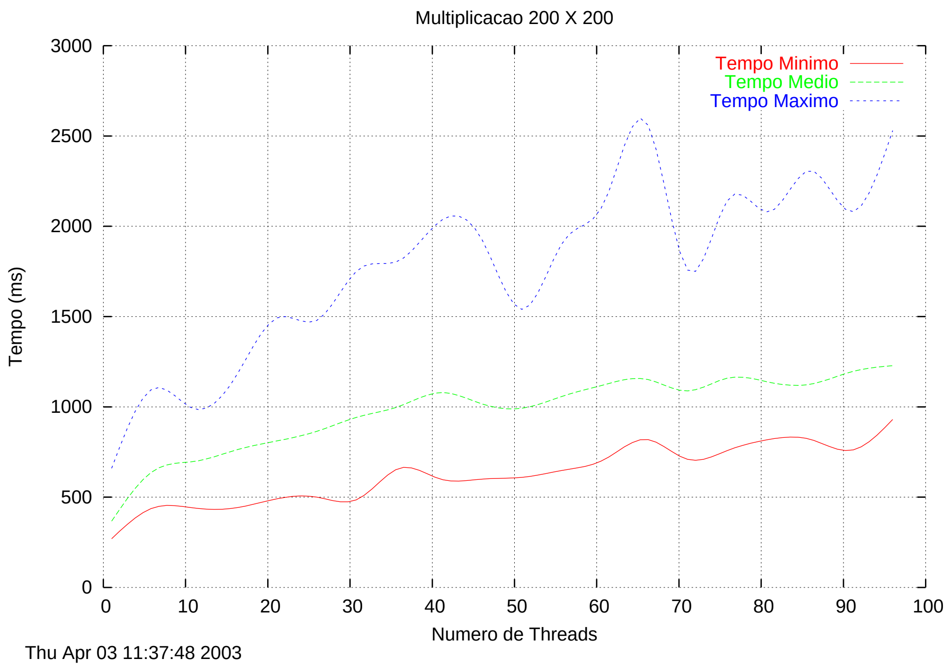
2.1.3 Teste

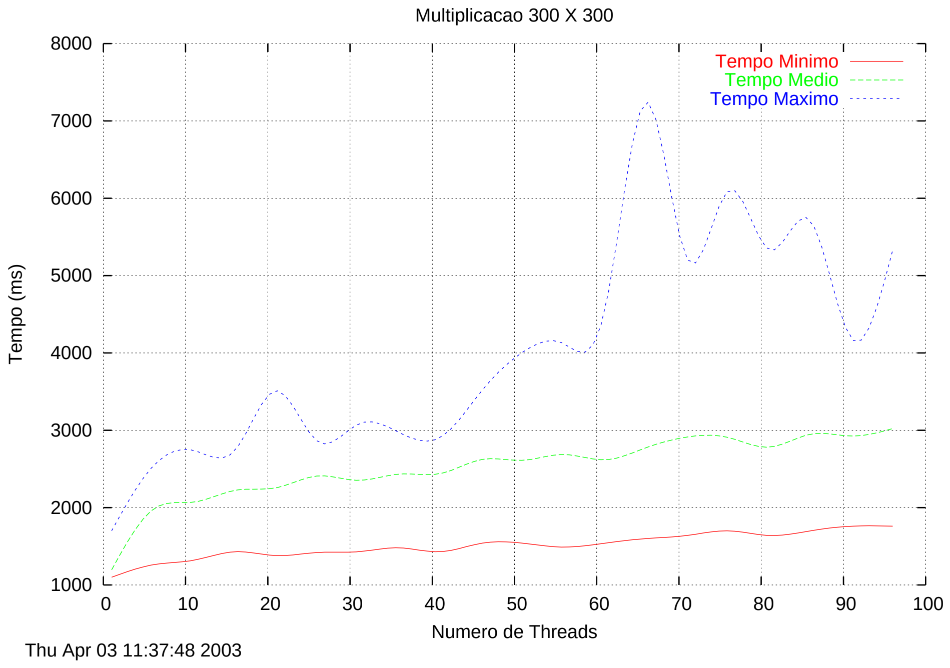
O teste foi realizado utilizando um script perl que efetuava chamadas sucessivas do nosso programa java. O script executava nosso programa cinqüenta vezes para cada quantidade de threads, estes eram increntados em intervalos de cinco threads, ou seja começavam em 1,6,11,...,96. As dimensões das matrizes eram incrementadas em intervalos de cem, ou seja inicialmente teriam cem linhas por cem colunas e eram aumentadas sucessivamentes até seiscentas linhas por seiscentas colunas.

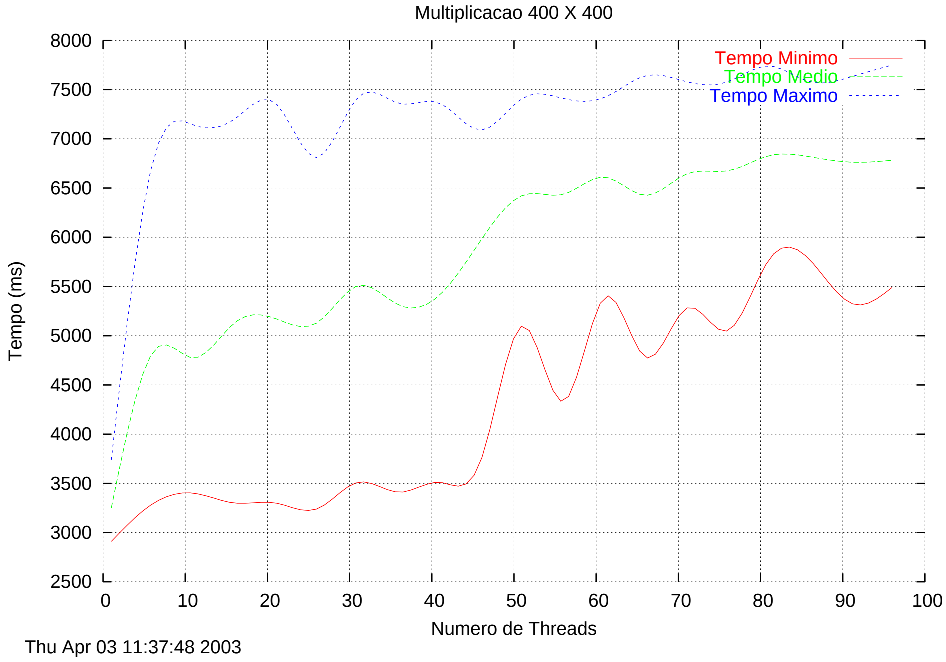
2.1.4 Gráficos

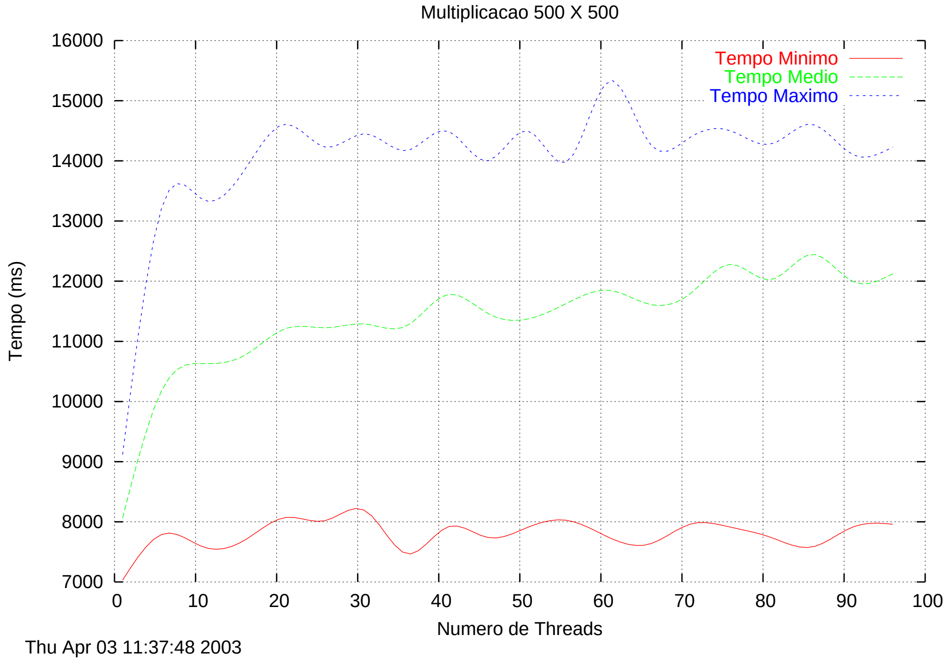
Foram gerados seis gráficos para essa máquina. Um gráfico para cada incremento nas dimensões das matrizes.

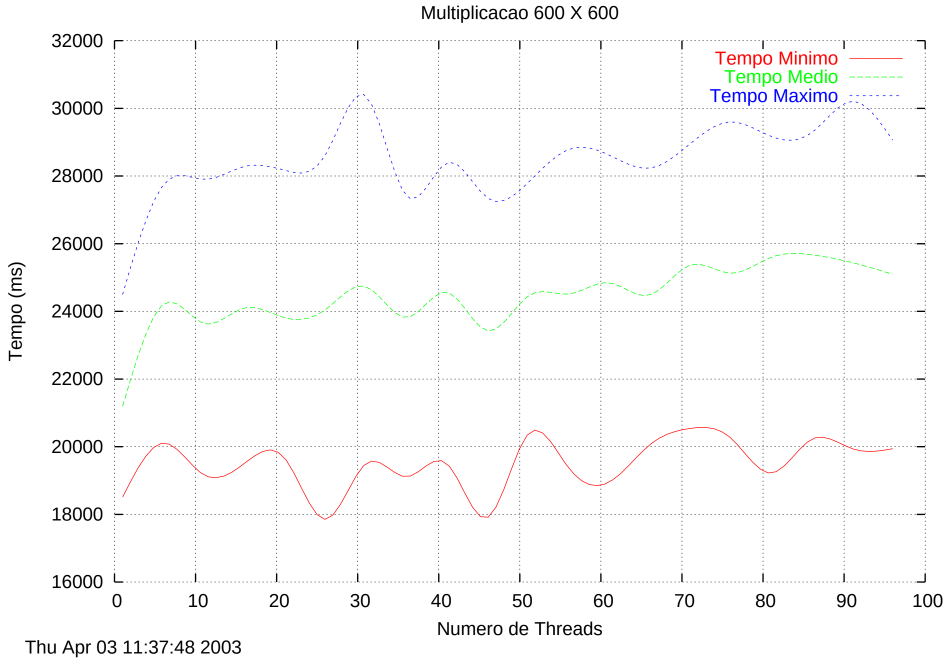



Multiplicacao 200 X 200
Tempo Minimo
Tempo Medio
Tempo Maximo
Tempo (ms)
3000
2500
2000
1500
1000
500
0
0
10
20
30
40
50
60
70
80
90
100
Numero de Threads
Thu Apr 03 11:37:48 2003


Multiplicacao 300 X 300
Tempo Minimo
Tempo Medio
Tempo Maximo
Tempo (ms)
8000
7000
6000
5000
4000
3000
2000
1000
0
10
20
30
40
50
60
70
80
90
100
Numero de Threads
Thu Apr 03 11:37:48 2003


Multiplicacao 400 X 400
Tempo Minimo
Tempo Medio
Tempo Maximo
Tempo (ms)
8000
7500
7000
6500
6000
5500
5000
4500
4000
3500
3000
2500
0
10
20
30
40
50
60
70
80
90
100
Numero de Threads
Thu Apr 03 11:37:48 2003


Multiplicacao 500 X 500
Tempo Minimo
Tempo Medio
Tempo Maximo
Tempo (ms)
16000
15000
14000
13000
12000
11000
10000
9000
8000
7000
0
10
20
30
40
50
60
70
80
90
100
Numero de Threads
Thu Apr 03 11:37:48 2003



2.2 Máquina Buba - Solaris

2.2.1 Local

A máquina Buba está localizada no bloco do Instituto de Matemática e Estátisca da Universidade de São Paulo à sala 259-B, onde se encontra o Laboratório de Banco de Dados - BIOINFO.

2.2.2 Configuração

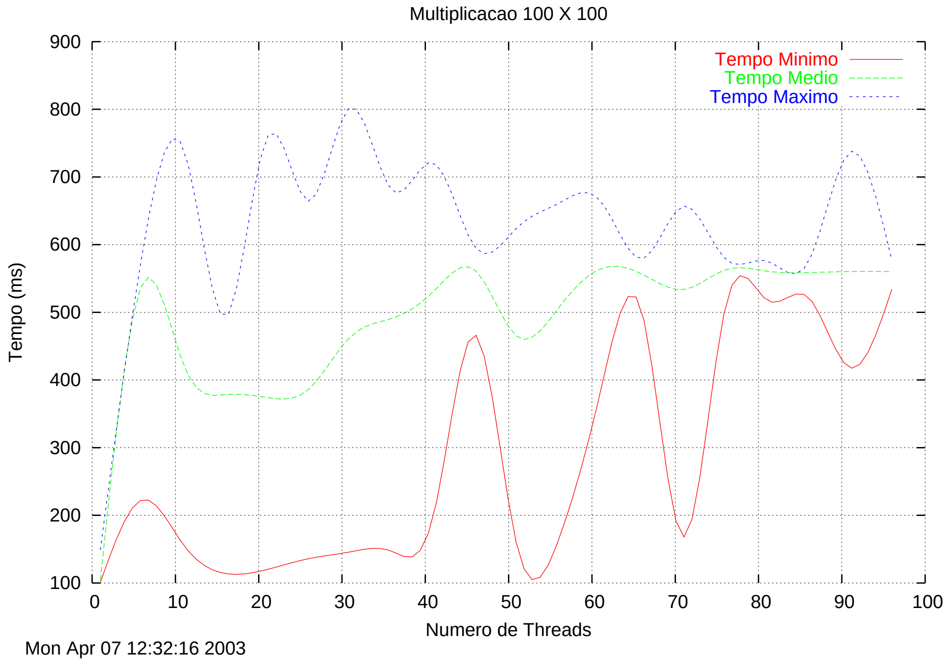
A máquina Buba possui um processador Sparc de 0.9 Ghz, 2 Gbytes de memória RAM, 40 Gbytes de hard disk e funciona sobre o sistema operacional Sun-Solaris 9 (o que é uma maravilha - em nossa opinião).

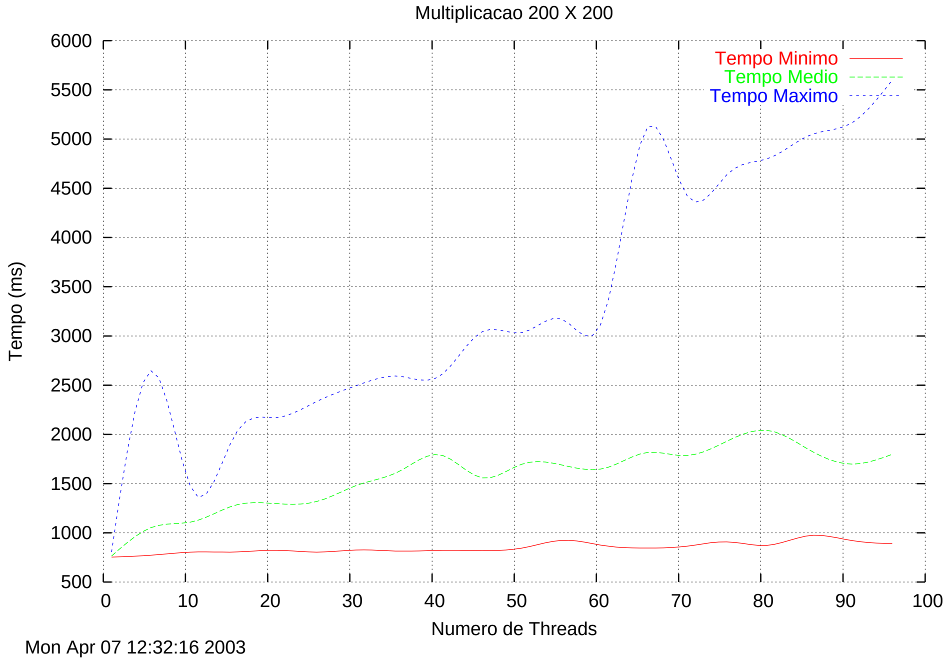
2.2.3 Teste

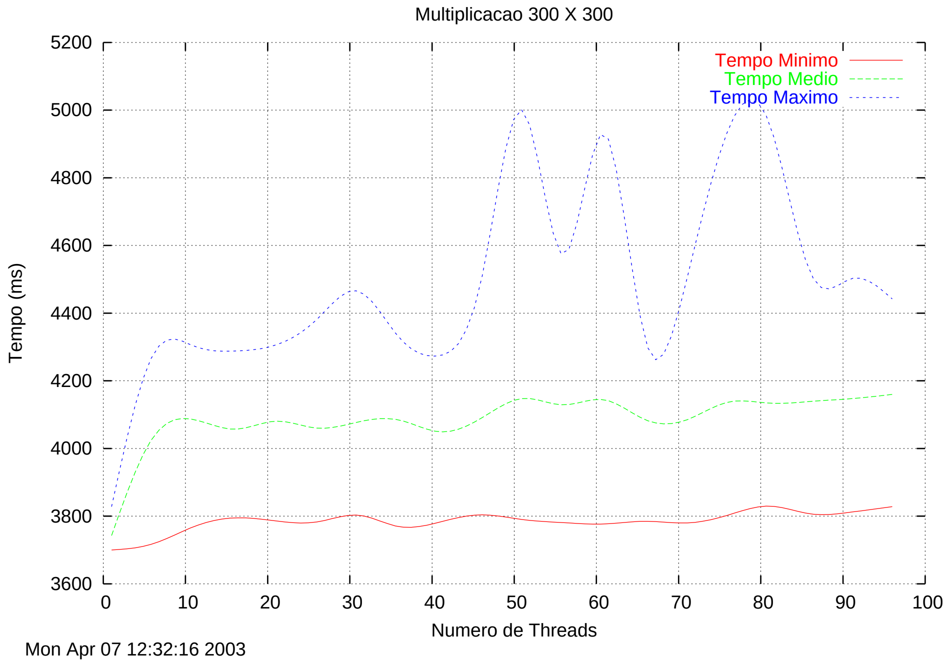
O teste foi realizado utilizando um script perl que efetuava chamadas sucessivas do nosso programa java. O script executava nosso programa cinqüenta vezes para cada quantidade de threads, estes eram increntados em intervalos de cinco threads, ou seja começavam em 1,6,11,...,96. As dimensões das matrizes eram incrementadas em intervalos de cem, ou seja inicialmente teriam cem linhas por cem colunas e eram aumentadas sucessivamentes até seiscentas linhas por seiscentas colunas.

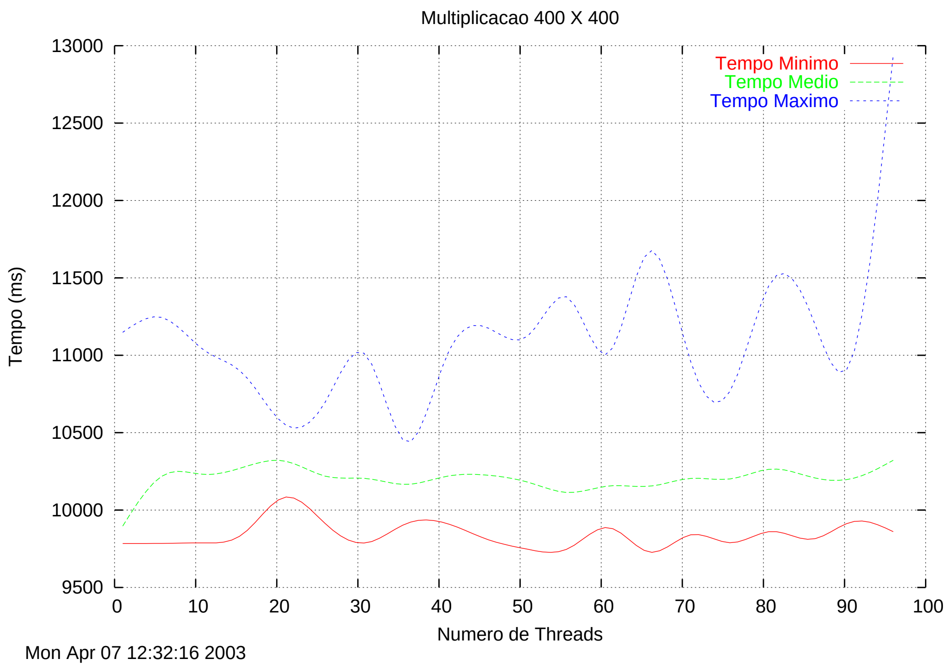
2.2.4 Gráficos

Foram gerados seis gráficos para essa máquina. Um gráfico para cada incremento nas dimensões das matrizes.

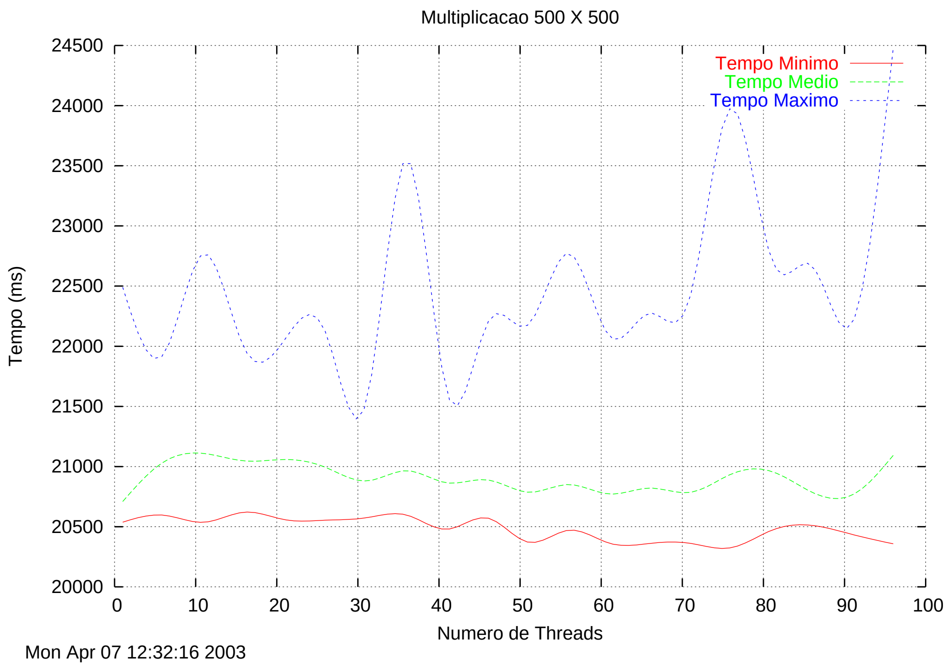



Multiplicacao 200 X 200
Tempo Minimo
Tempo Medio
Tempo Maximo
Tempo (ms)
6000
5500
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
10
20
30
40
50
60
70
80
90
100
Numero de Threads
Mon Apr 07 12:32:16 2003

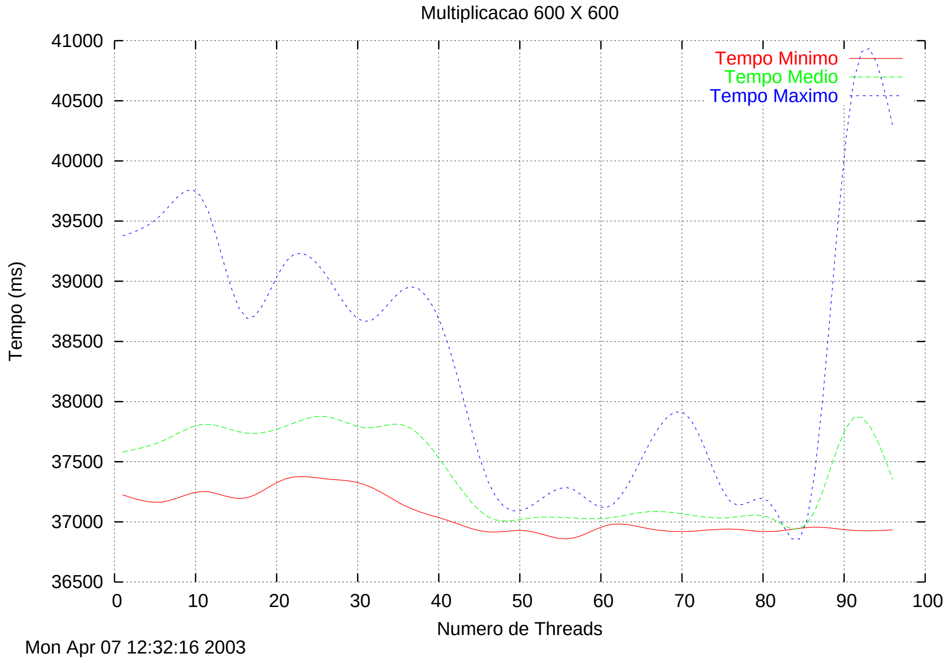

Multiplicacao 300 X 300
Tempo Minimo
Tempo Medio
Tempo Maximo
Tempo (ms)
5200
5000
4800
4600
4400
4200
4000
3800
3600
0
10
20
30
40
50
60
70
80
90
100
Numero de Threads
Mon Apr 07 12:32:16 2003



Multiplicacao 400 X 400
Tempo Minimo
Tempo Medio
Tempo Maximo
Tempo (ms)
13000
12500
12000
11500
11000
10500
10000
9500
0
10
20
30
40
50
60
70
80
90
100
Numero de Threads
Mon Apr 07 12:32:16 2003



Multiplicacao 500 X 500
Tempo Minimo
Tempo Medio
Tempo Maximo
Tempo (ms)
24500
24000
23500
23000
22500
22000
21500
21000
20500
20000
0
10
20
30
40
50
60
70
80
90
100
Numero de Threads
Mon Apr 07 12:32:16 2003



2.3 Máquina Cage - Solaris

2.3.1 Local

A máquina Cage está localizada no bloco do Instituto de Matemática e Estátisca da Universidade de São Paulo à sala 259-B, onde se encontra o Laboratório de Banco de Dados - BIOINFO.

2.3.2 Configuração

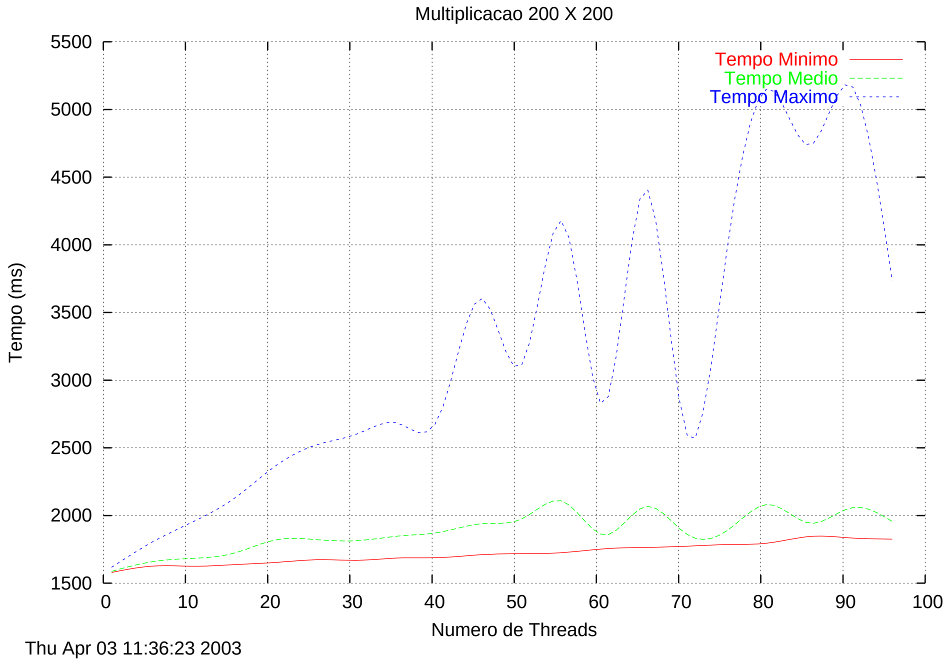
A máquina Cage possui um processador Sparc de 0.4 Ghz, 256 Mb de memória RAM, 40 Gbytes de hard disk e funciona sobre o sistema operacional Windows 98 (o que é uma beleza - em nossa opinião).

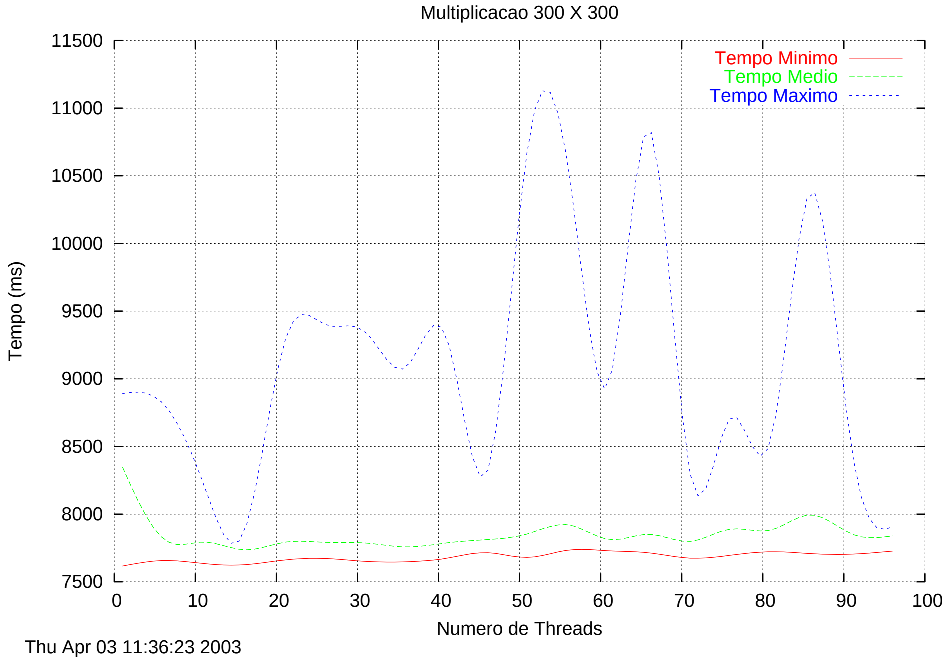
2.3.3 Teste

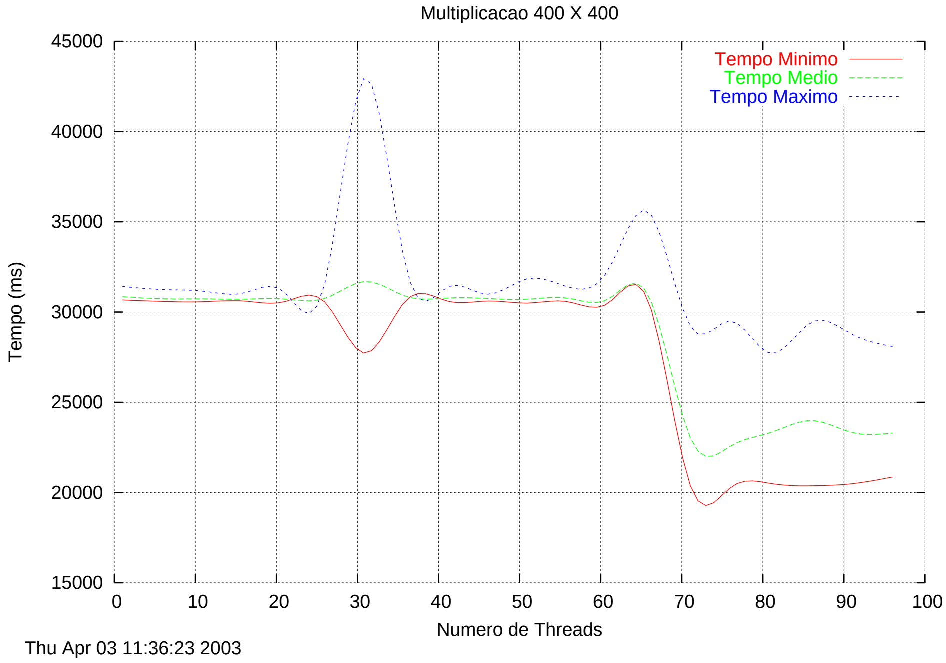
O teste foi realizado utilizando um script perl que efetuava chamadas sucessivas do nosso programa java. O script executava nosso programa cinqüenta vezes para cada quantidade de threads, estes eram increntados em intervalos de cinco threads, ou seja começavam em 1,6,11,...,96. As dimensões das matrizes eram incrementadas em intervalos de cem, ou seja inicialmente teriam cem linhas por cem colunas e eram aumentadas sucessivamentes até seiscentas linhas por seiscentas colunas.

2.3.4 Gráficos

Foram gerados seis gráficos para essa máquina. Um gráfico para cada incremento nas dimensões das matrizes.




Multiplicacao 300 X 300
Tempo Minimo
Tempo Medio
Tempo Maximo
Tempo (ms)
Numero de Threads
11500
11000
10500
10000
9500
9000
8500
8000
7500
0
10
20
30
40
50
60
70
80
90
100
Thu Apr 03 11:36:23 2003


Multiplicacao 400 X 400
Tempo Minimo
Tempo Medio
Tempo Maximo
Tempo (ms)
Numero de Threads
45000
40000
35000
30000
25000
20000
15000
0
10
20
30
40
50
60
70
80
90
100
Thu Apr 03 11:36:23 2003

3 Máquinas multiprocessadas

Os testes em máquinas processadas foram realizadas em dois servidos, ambos com o sistema operecional Sun-Solaris 9.

3.1 Servidor - Cagedb

3.1.1 Local

A máquina Cage está localizada no bloco do Instituto de Matemática e Estátisca da Universidade de São Paulo o local é secreto, mas este também e de propriedade do BIOINFO.

3.1.2 Configuração

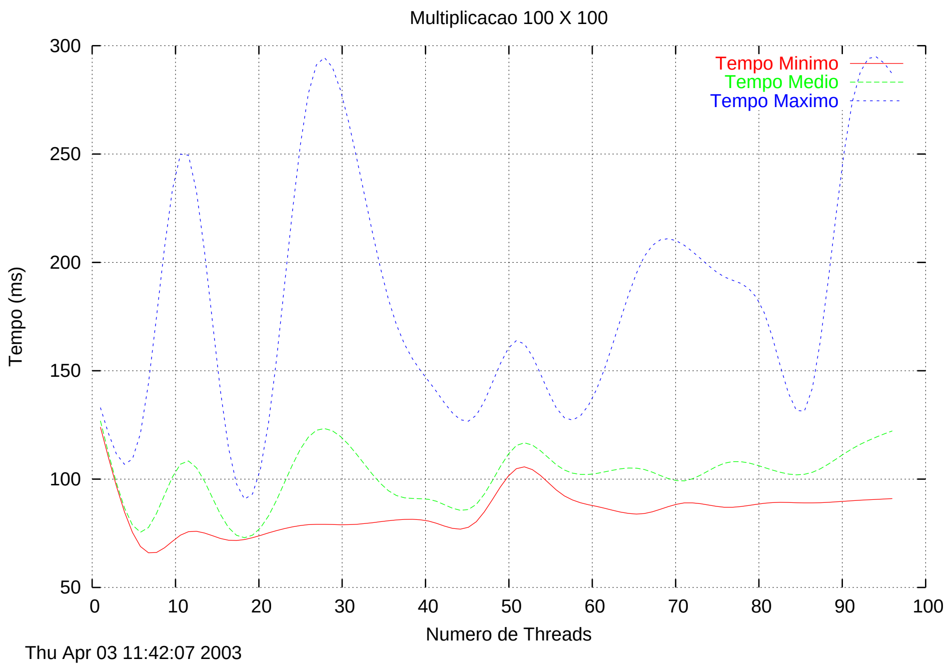
O servidor Cagedb possui dois processadores Sparc de ??? Ghz, 2 Gbytes de memória RAM, 40 Gbytes de hard disk.

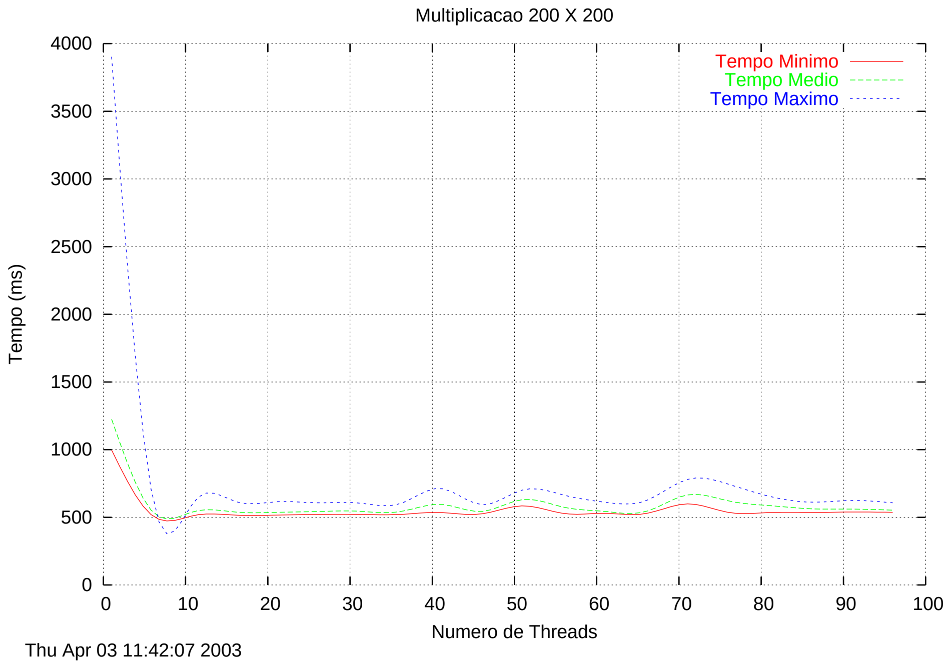
3.1.3 Teste

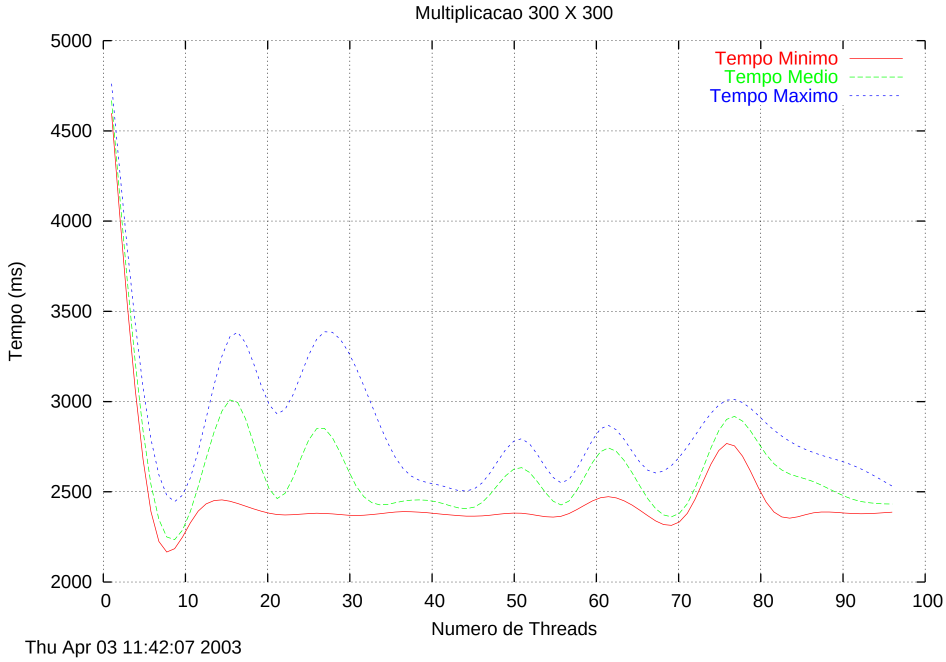
O teste foi realizado utilizando um script perl que efetuava chamadas sucessivas do nosso programa java. O script executava nosso programa cinqüenta vezes para cada quantidade de threads, estes eram increntados em intervalos de cinco threads, ou seja começavam em 1,6,11,...,96. As dimensões das matrizes eram incrementadas em intervalos de cem, ou seja inicialmente teriam cem linhas por cem colunas e eram aumentadas sucessivamentes até seiscentas linhas por seiscentas colunas.

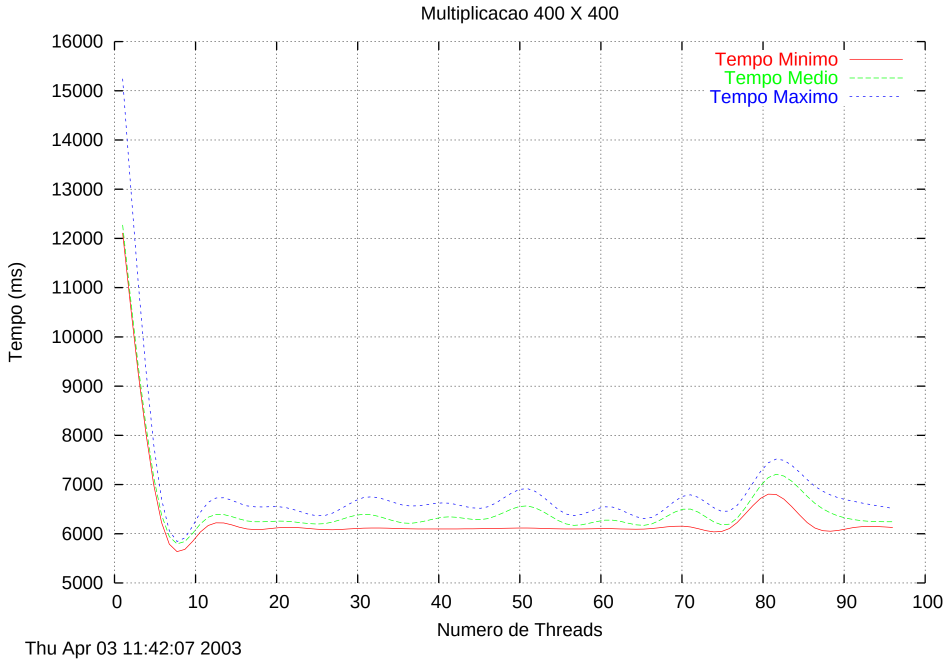
3.1.4 Gráficos

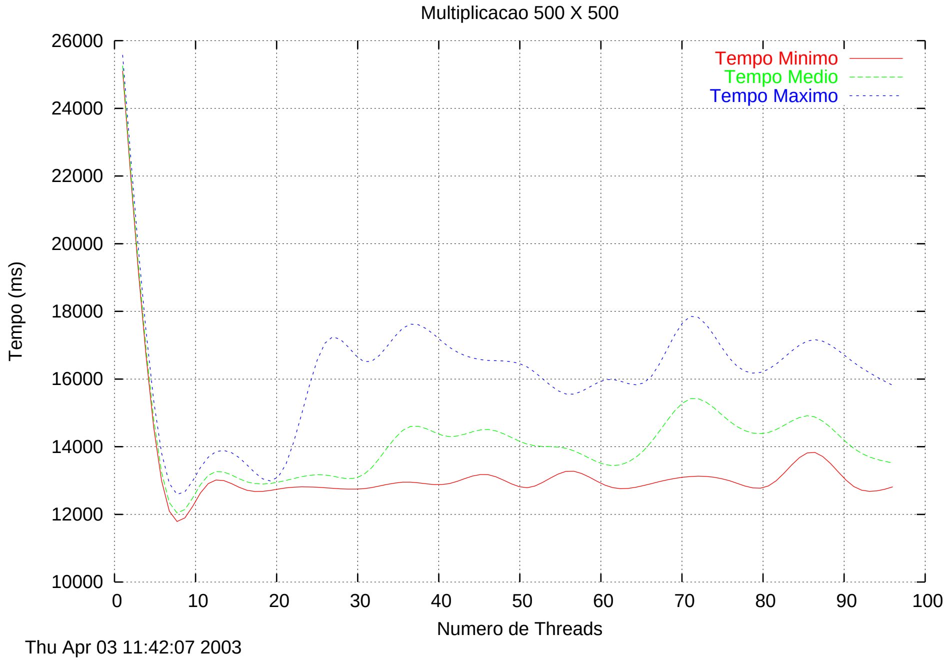
Foram gerados seis gráficos para essa máquina. Um gráfico para cada incremento nas dimensões das matrizes.

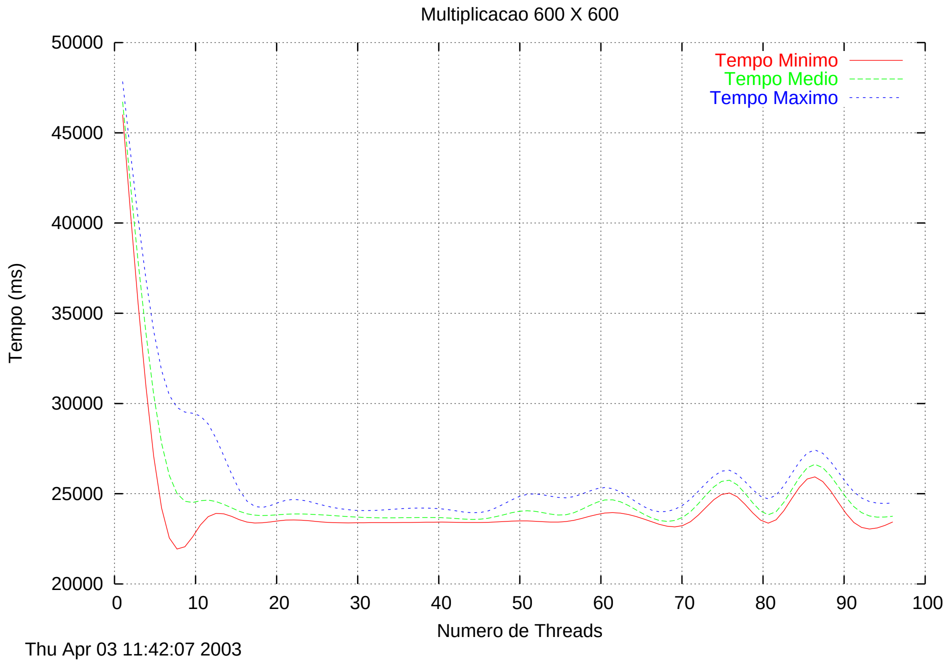

Multiplicacao 100 X 100
Tempo Minimo
Tempo Medio
Tempo Maximo
Tempo (ms)
300
250
200
150
100
50
0
10
20
30
40
50
60
70
80
90
100
Numero de Threads
Thu Apr 03 11:42:07 2003


Multiplicacao 200 X 200
Tempo Minimo
Tempo Medio
Tempo Maximo
Tempo (ms)
4000
3500
3000
2500
2000
1500
1000
500
0
10
20
30
40
50
60
70
80
90
100
Numero de Threads
Thu Apr 03 11:42:07 2003


Multiplicacao 300 X 300
Tempo Minimo
Tempo Medio
Tempo Maximo
Tempo (ms)
Numero de Threads
5000
4500
4000
3500
3000
2500
2000
0
10
20
30
40
50
60
70
80
90
100
Thu Apr 03 11:42:07 2003


Multiplicacao 400 X 400
Tempo Minimo
Tempo Medio
Tempo Maximo
Tempo (ms)
Numero de Threads
16000
15000
14000
13000
12000
11000
10000
9000
8000
7000
6000
5000
0
10
20
30
40
50
60
70
80
90
100
Thu Apr 03 11:42:07 2003


Multiplicacao 500 X 500
Tempo Minimo
Tempo Medio
Tempo Maximo
Tempo (ms)
26000
24000
22000
20000
18000
16000
14000
12000
10000
0
10
20
30
40
50
60
70
80
90
100
Numero de Threads
Thu Apr 03 11:42:07 2003


Multiplicacao 600 X 600
Tempo Minimo
Tempo Medio
Tempo Maximo
Tempo (ms)
50000
45000
40000
35000
30000
25000
20000
0
10
20
30
40
50
60
70
80
90
100
Numero de Threads
Thu Apr 03 11:42:07 2003

3.2 Servidor - Malariadb

3.2.1 Local

A máquina Cage está localizada no bloco do Instituto de Matemática e Estátisca da Universidade de São Paulo o local é secreto, mas este também e de propriedade do BIOINFO.

3.2.2 Configuração

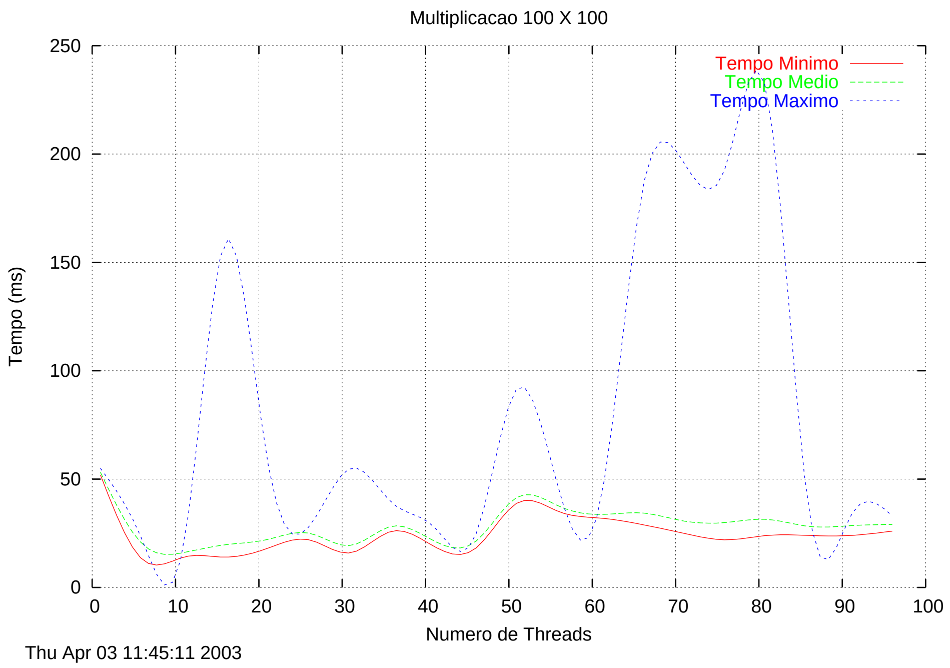
O servidor Cagedb possui oito processadores Sparc de 0.9 Ghz, 32 Gbytes de memória RAM, muitos Gbytes de hard disk.

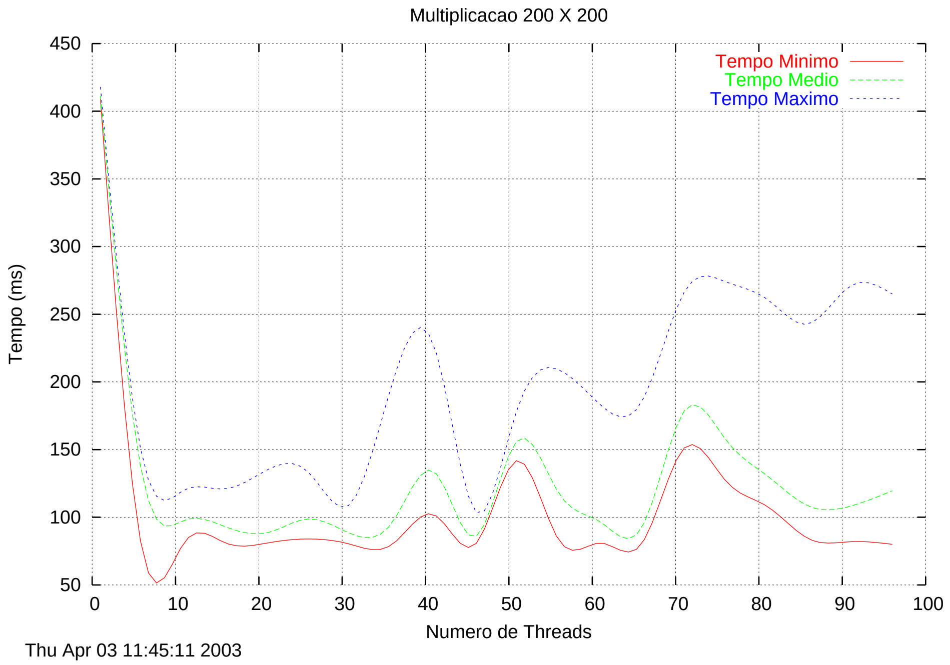
3.2.3 Teste

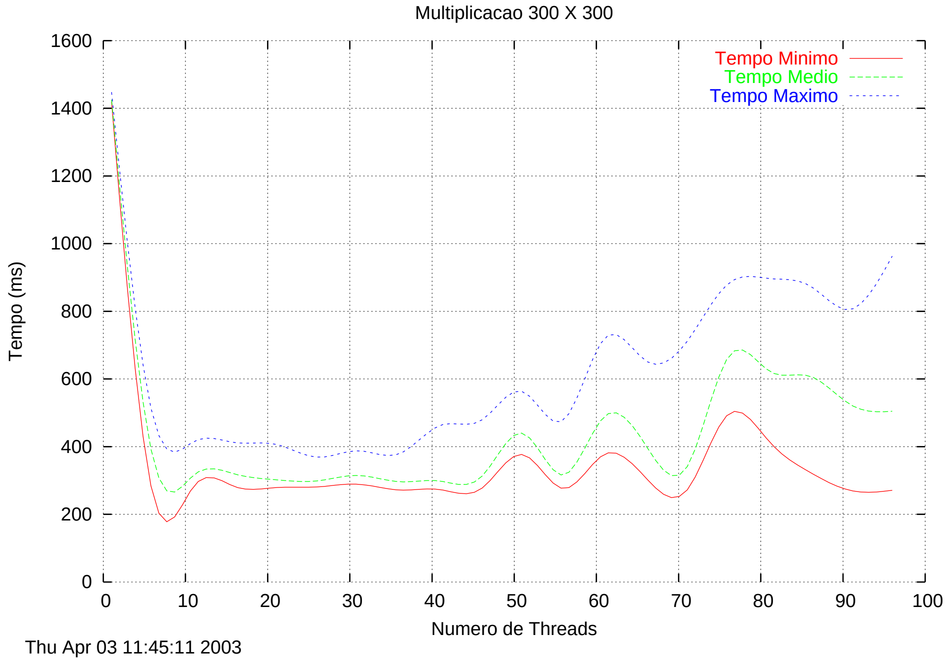
O teste foi realizado utilizando um script perl que efetuava chamadas sucessivas do nosso programa java. O script executava nosso programa cinqüenta vezes para cada quantidade de threads, estes eram increntados em intervalos de cinco threads, ou seja começavam em 1,6,11,...,96. As dimensões das matrizes eram incrementadas em intervalos de cem, ou seja inicialmente teriam cem linhas por cem colunas e eram aumentadas sucessivamentes até seiscentas linhas por seiscentas colunas.

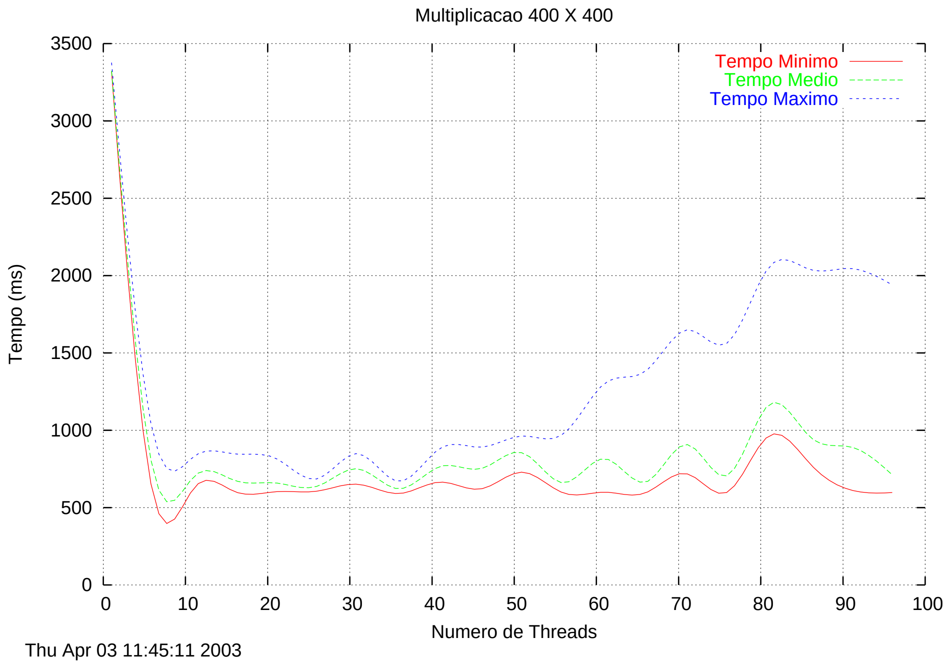
3.2.4 Gráficos

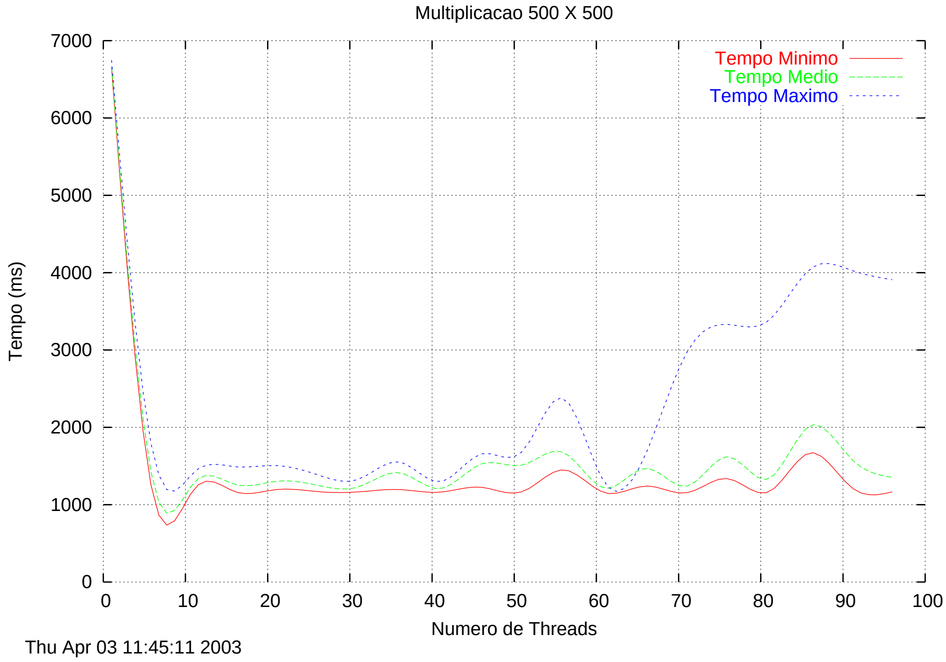
Foram gerados seis gráficos para essa máquina. Um gráfico para cada incremento nas dimensões das matrizes.

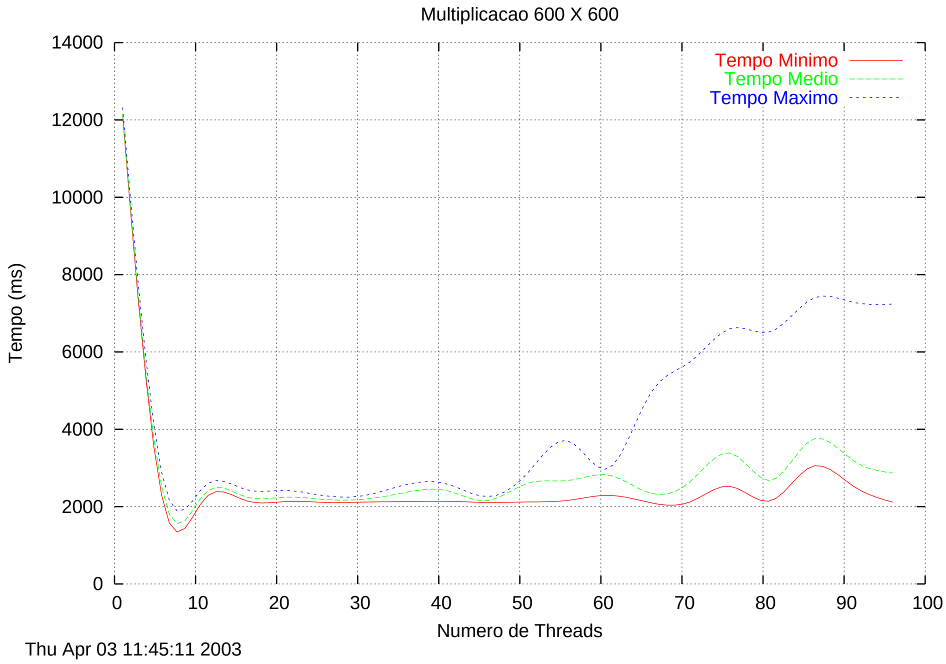

Multiplicacao 100 X 100
Tempo (ms)
Numero de Threads
Tempo Minimo
Tempo Medio
Tempo Maximo
Thu Apr 03 11:45:11 2003

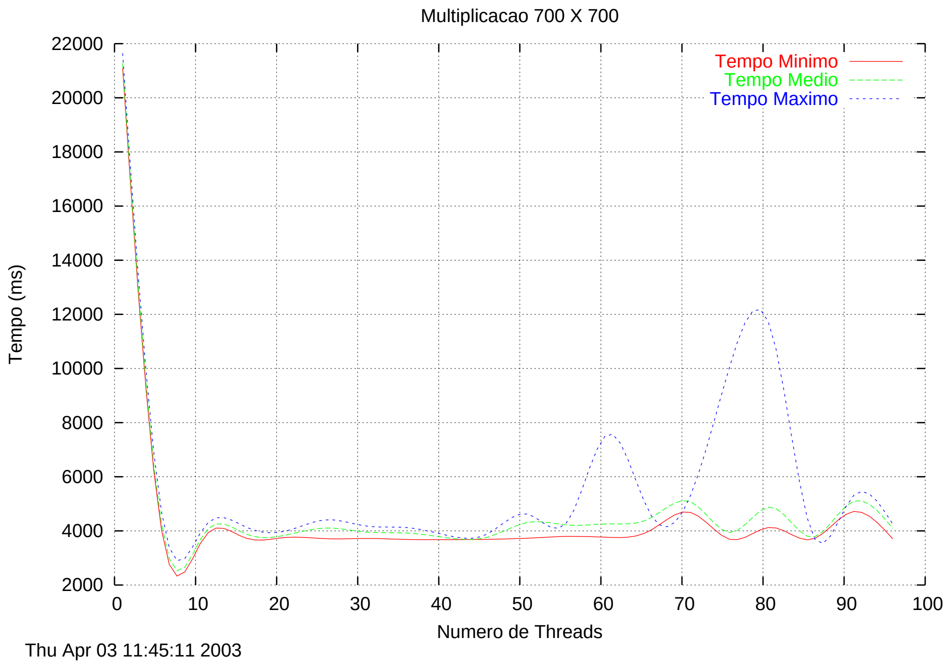

Multiplicacao 200 X 200
Tempo (ms)
Numero de Threads
Tempo Minimo
Tempo Medio
Tempo Maximo
Thu Apr 03 11:45:11 2003

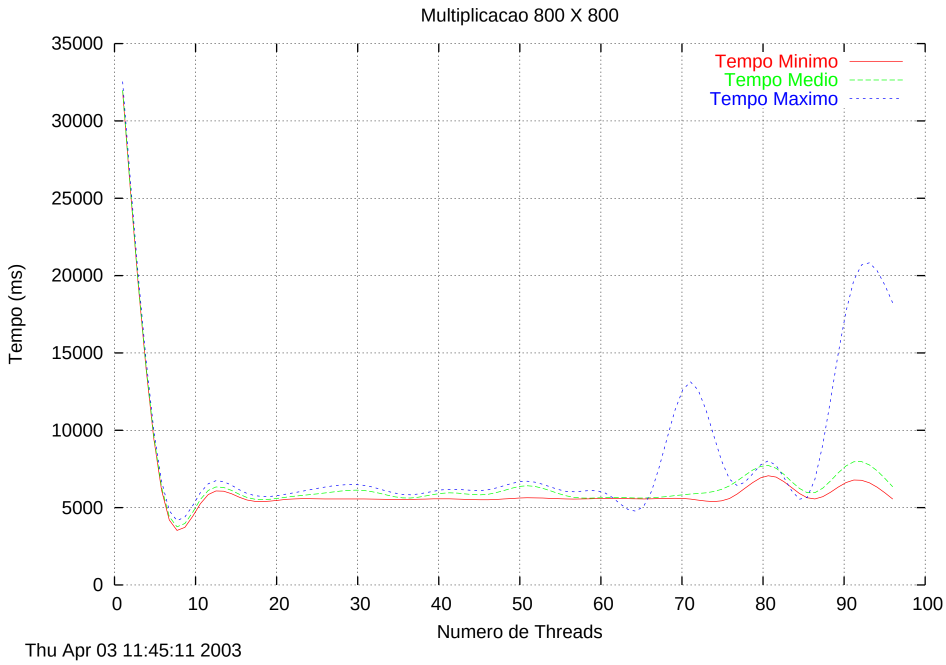

Multiplicacao 300 X 300
Tempo Minimo
Tempo Medio
Tempo Maximo
Tempo (ms)
0
200
400
600
800
1000
1200
1400
1600
0
10
20
30
40
50
60
70
80
90
100
Numero de Threads
Thu Apr 03 11:45:11 2003

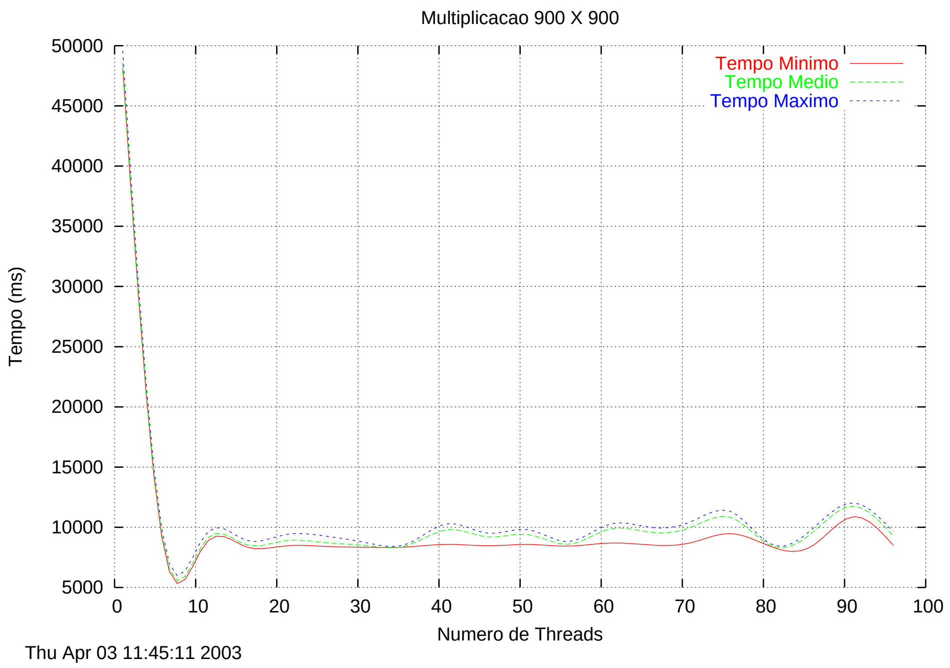

Multiplicacao 400 X 400
Tempo Minimo
Tempo Medio
Tempo Maximo
Tempo (ms)
0
500
1000
1500
2000
2500
3000
3500
0
10
20
30
40
50
60
70
80
90
100
Numero de Threads
Thu Apr 03 11:45:11 2003

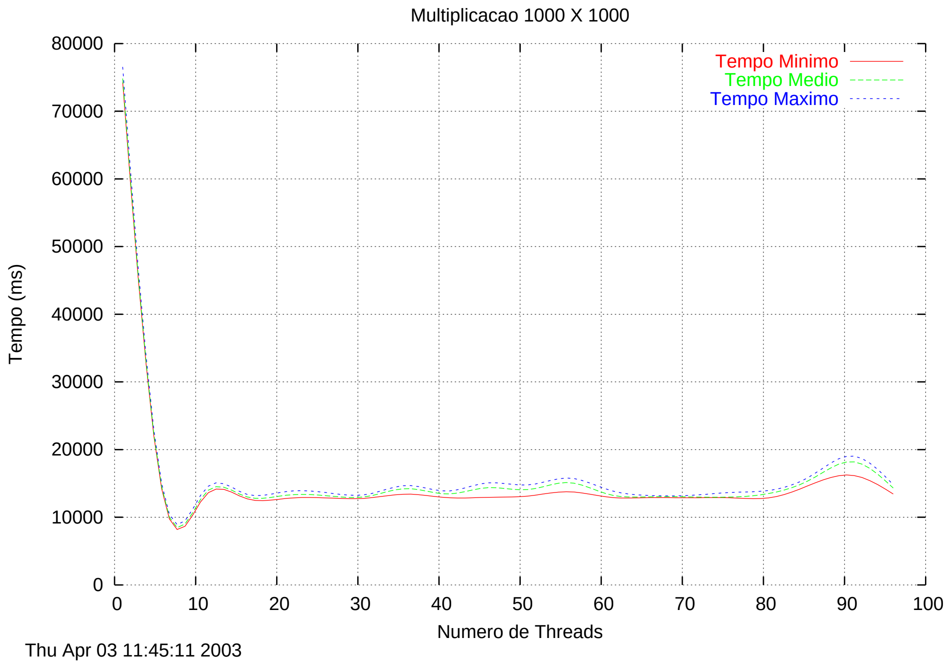

Multiplicacao 500 X 500
Tempo Minimo
Tempo Medio
Tempo Maximo
Tempo (ms)
0
1000
2000
3000
4000
5000
6000
7000
0 10 20 30 40 50 60 70 80 90 100
Numero de Threads
Thu Apr 03 11:45:11 2003


Multiplicacao 600 X 600
Tempo Minimo
Tempo Medio
Tempo Maximo
Tempo (ms)
0
2000
4000
6000
8000
10000
12000
14000
0 10 20 30 40 50 60 70 80 90 100
Numero de Threads
Thu Apr 03 11:45:11 2003


Multiplicacao 700 X 700
Tempo Minimo
Tempo Medio
Tempo Maximo
Tempo (ms)
22000
20000
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
10
20
30
40
50
60
70
80
90
100
Numero de Threads
Thu Apr 03 11:45:11 2003


Multiplicacao 800 X 800
Tempo Minimo
Tempo Medio
Tempo Maximo
Tempo (ms)
35000
30000
25000
20000
15000
10000
5000
0
0
10
20
30
40
50
60
70
80
90
100
Numero de Threads
Thu Apr 03 11:45:11 2003


Multiplicacao 900 X 900
Tempo Minimo
Tempo Medio
Tempo Maximo
50000
45000
40000
35000
30000
25000
20000
15000
10000
5000
Tempo (ms)
0
10
20
30
40
50
60
70
80
90
100
Numero de Threads
Thu Apr 03 11:45:11 2003


Multiplicacao 1000 X 1000
Tempo Minimo
Tempo Medio
Tempo Maximo
80000
70000
60000
50000
40000
30000
20000
10000
0
Tempo (ms)
0
10
20
30
40
50
60
70
80
90
100
Numero de Threads
Thu Apr 03 11:45:11 2003

4 Conclusão

4.1 Monoprocessamento

Podemos observar em cada uma dos gráficos que foram gerados por máquinas monoprocessadas um comportamento crescente para cada função gerada. Isso tem uma certa lógica, pois conforme o número de *threads* vai aumentando temos um maior *overhead*. Isto devido a constante troca de contexto da CPU aumentando o tempo de execução do programa java. Quanto maior o número de trocas mais se perde com o tempo de latência, ou seja, mais se perde tempo para gravar as informações do processo que está execução para que a troca seja realizada.

Nós observamos nos gráficos gerados pela execução das máquinas monoprocessadas um comportamento meio aleatório (mas mesmo com essa aleatoriedade temos ainda uma predominância do crescimento). Tal comportamento foi ocasionado pelo fato de que o tempo computado para cada execução foi o tempo real de execução e não o tempo de uso de CPU (devemos destacar ainda que as máquinas continuaram em execução normal durante as execuções, ou seja, haviam pessoas trabalhando nelas, o que poderia explicar a ocorrência de picos de tempo que geraram grandes oscilações nos gráficos), mas o comportamento de cada função não foi alterado em seu todo.

4.2 Multiprocessamento

Ao contrário do que acontece com a máquinas monoprocessadas observamos em máquinas multiprocessadas um comportamento predominantemente decrescente quanto ao tempo de execução dos testes. À medida que o número de threads era aumentado tinhamos uma pequena queda no tempo de execução, mas quando este número ultrapassava o número total de procesadores obtinhamos uma certa estabilidade em relação ao tempo de execução, isto se deve ao aumento de trocas de contexto.