

CAP-387(2016) – Tópicos Especiais em Computação Aplicada: Construção de Aplicações Massivamente Paralelas

Aula 8: Medição de Tempo

Celso L. Mendes, Stephan Stephany

LAC /INPE

Emails: celso.mendes@inpe.br, stephan.stephany@inpe.br



Características de Relógios

- **Resolução:**
 - Menor tempo que pode ser medido com o relógio
(na prática: tempo que deve decorrer para que o relógio mude de valor)
 - Também conhecida como *clock-tick*
- **Erro de Medição:**
 - Da ordem de um clock-tick
 - Exemplo: imagine um relógio sem ponteiro de segundos
 - Resolução = 1 minuto
 - Apenas durações em minutos podem ser medidas
 - Se duração real = 6'59" → valor medido = 6 minutos! Erro=59/419=14%
 - Idealmente: trecho a ser medido deve ter duração maior que ~100 ticks
 - Erro de medição $\leq 1\%$



Tipos de Relógios

- **gettimeofday():**
 - Portável para praticamente todo Sistema Unix/Linux
 - Retorna tempo (em dois valores, *seconds* : *microseconds*) decorrido desde um certo momento no passado → resolução = 1 μ s
 - Não é estritamente crescente, mas suficiente para execuções de interesse
- **Para este curso: *double mysecond()***
 - Retorna o valor atual do relógio, em segundos
 - Baseado em *gettimeofday()*: (double)sec + 10^{-6} * (double) μ sec
- **Outros relógios:**
 - Possivelmente melhor resolução, porém não são portáveis



Tipos de Relógios (cont.)

- **Programas em OpenMP:**
 - Funções *omp_get_wtime()*, *omp_get_wtick()*
 - *wtime*: lê o relógio; *wtick*: retorna o *clock-tick* do Sistema
 - Ambas retornam valores em *segundos*
 - Exemplo: *double t = omp_get_wtime()*
 - Especificadas no padrão OpenMP
- **Programas em MPI:**
 - Funções *MPI_Wtime()*, *MPI_Wtick()*
 - *Wtime*: lê o relógio; *Wtick*: retorna o *clock-tick* do Sistema
 - Ambas retornam valores em *segundos*
 - Exemplo: *double t = MPI_Wtime()*
 - Especificadas no padrão MPI



Caracterização de Relógios

- Possível código de avaliação:

$T_0 = le_relogio()$

$for (i=0; i<N; i++) T(i) = le_relogio();$

$delta = T(N-1) - T_0$

- Custo da chamada ao relógio: $delta/N$
- Resolução do relógio: Intervalo de tempo decorrido tal que $T(i) \neq T(i+k)$ para o menor k possível

Exemplo de Caracterização

```
double mysecond(void);

int main(int argc, char *argv[]) {
    double *ticks, mindist, maxdist, dist;
    int i, n=10000, maxn;
    if (argc > 1) n = atoi(argv[1]);
    ticks = (double *)malloc(n * sizeof(double));
    for (i=0; i<n; i++) { ticks[i] = mysecond(); }
    mindist = 1e10;
    maxdist = 0;
    for (i=1; i<n; i++) {
        dist = ticks[i] - ticks[i-1];
        if (dist > 0) {
            if (dist < mindist) mindist = dist;
            if (dist > maxdist) maxdist = dist;
        }
    }
    printf("# min dist = %.2e, max dist = %.2e, total time = %.2e\n",
        mindist, maxdist, ticks[n-1] - ticks[0]);
    for (i=1; i<n; i++) { printf("%d\t%.2e\n", i, ticks[i] - ticks[i-1]); }
}
```



Exemplo no Santos Dumont

- Saída gerada pela execução em um nó (n=10.000):

min dist = 9.54e-07, max dist = 2.15e-06, total time = 4.64e-04

1 9.54e-07

2 0.00e+00

3 0.00e+00

4 0.00e+00

...

20 0.00e+00

21 0.00e+00

22 9.54e-07

23 0.00e+00

24 0.00e+00

...

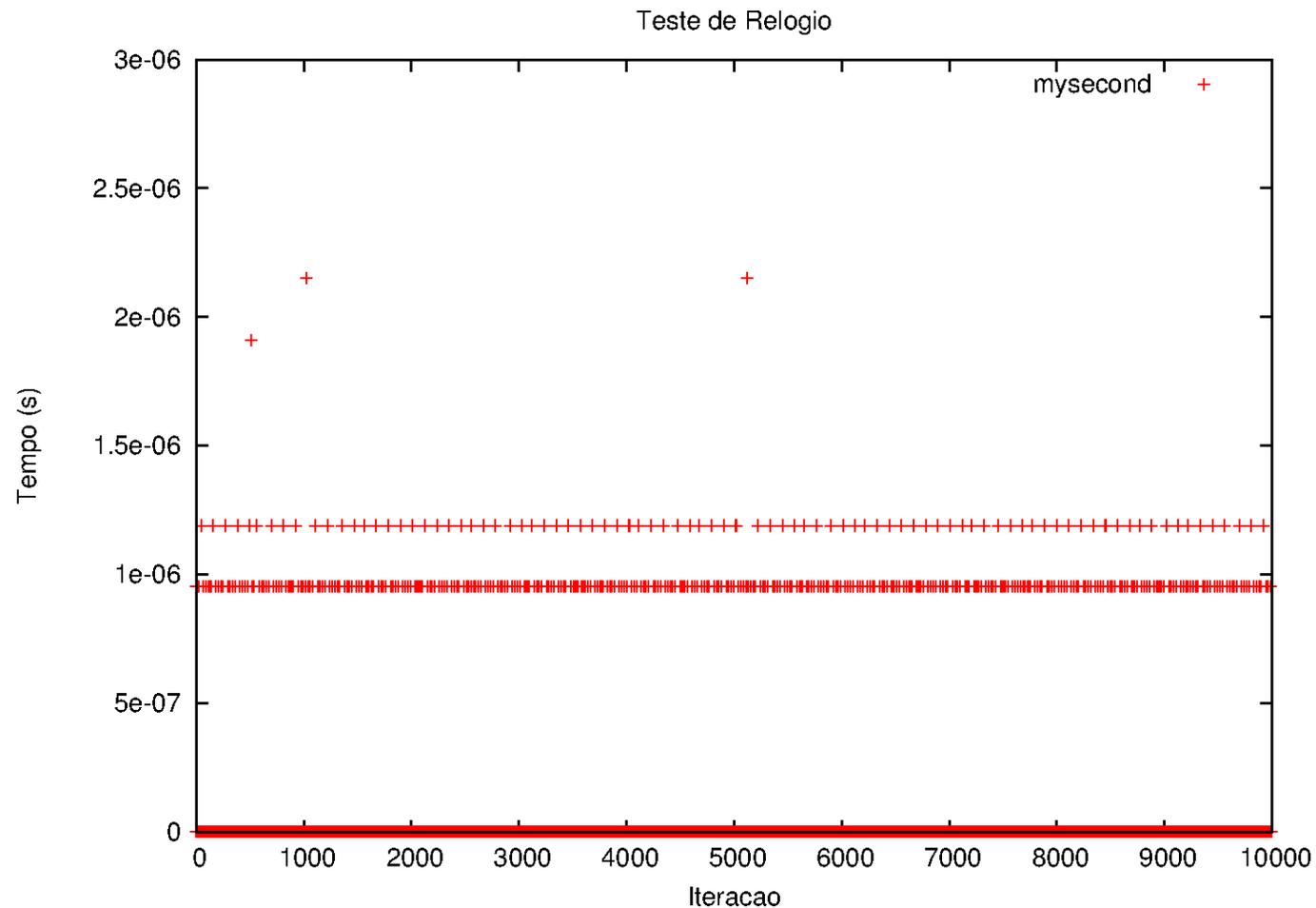
Delta = 464 μ s , n=10.000

Logo, custo por chamada \approx 46 ns

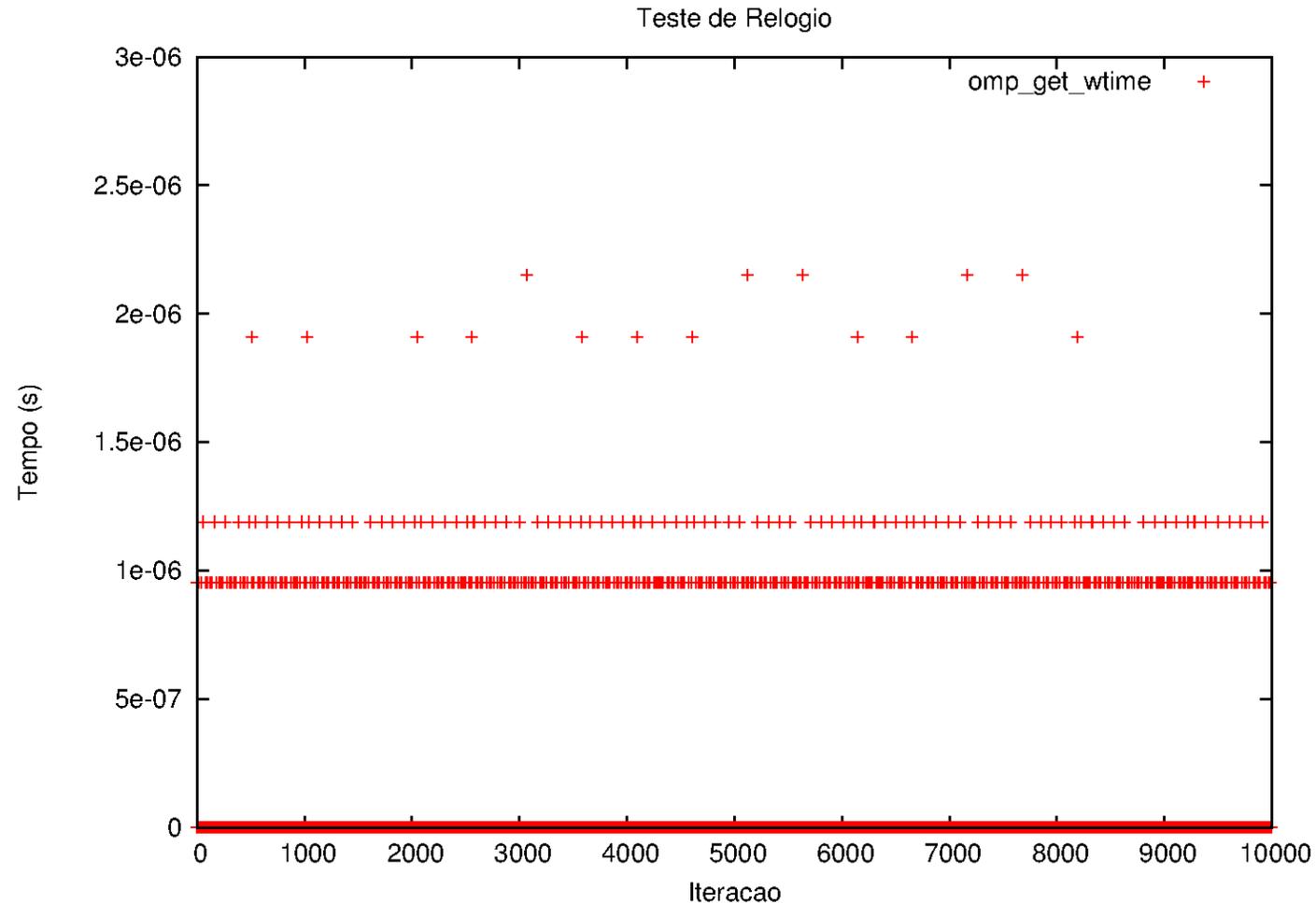
$T(22)-T(1) \approx (22-1)*46 \text{ ns} = 966 \text{ ns} \approx \text{resolução} = 1 \mu\text{s}$



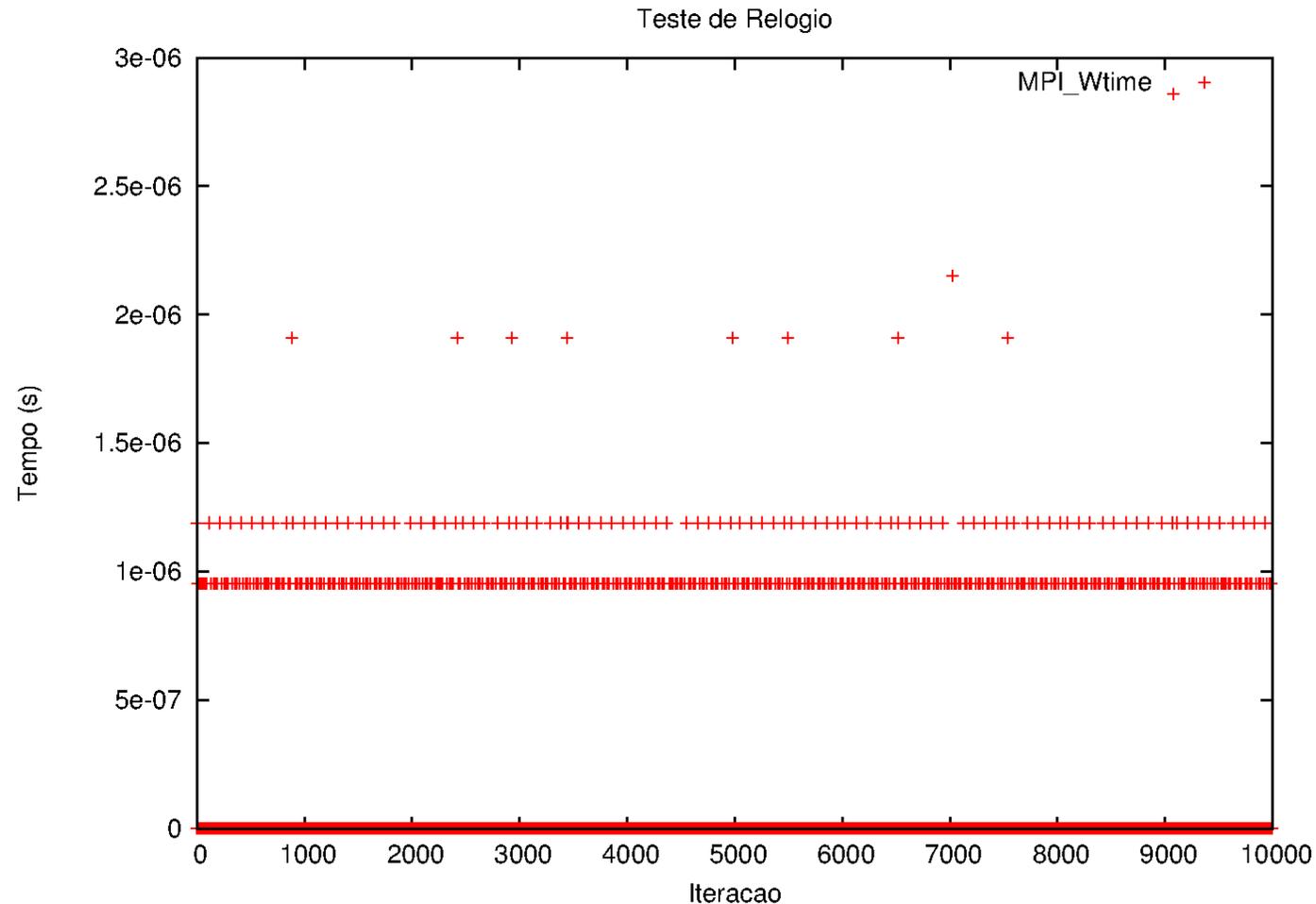
Exemplo no Santos Dumont



Exemplo com Relógio OpenMP



Exemplo com Relógio MPI



Resultados no Santos Dumont

- $n=10^4$:

Relógio	Mín	Máx	Delta
mysecond	0,954 μ s	2,15 μ s	464 μ s
omp_get_wtime	0,954 μ s	5,96 μ s	533 μ s
MPI_Wtime	0,954 μ s	12,2 μ s	555 μ s

- $n=10^6$:

Relógio	Mín	Máx	Delta
mysecond	0,954 μ s	11,9 μ s	48,3 ms
omp_get_wtime	0,954 μ s	11,0 μ s	52,9 ms
MPI_Wtime	0,954 μ s	16,9 μ s	53,0 ms

