

Universidade do Minho
Ano letivo 2016/2017
Mestrado Integrado em Engenharia Informática

Paradigmas de Computação Paralela

Simulação do Processo de Difusão de Calor

Grupo 3
João Lopes a61077, Nuno Moreira a61017

Departamento de Informática
Escola de Engenharia



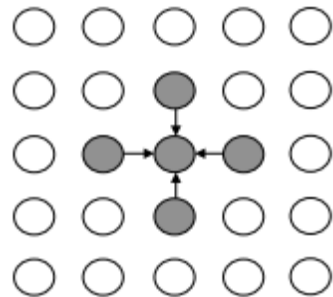
Estrutura da Apresentação

- Caso de estudo
- Solução sequencial
- Solução paralela – OpenMP
- SpeedUp's – OpenMP
- Solução paralela – OpenMPI
- Testes/medições e conclusões – OpenMPI



Caso de Estudo:

- Simulação do processo de difusão de calor em N_MAX iterações numa matriz quadrada representativa da superfície
- Periferias da matriz permanecem imutáveis
- Calculo da temperatura num determinado ponto segue a forma de um algoritmo *stencil*



100	100	100	100	100	100
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Solução Sequencial

- Tamanho do input:
 - Matrizes de *floats* de tamanhos 1024x1024, 2048x2048, 4096x4096 e 8192x8192
 - Matriz 1024x1024 consegue ser totalmente inserida na memória cache L3
- N_MAX iterações:
 - 1000, 2000, 4000 e 8000



Solução Paralela - OpenMP

```
int i, j;

//Itera ate N_MAX iteracoes
for(int iteration = 0; iteration < N_MAX; iteration++) {
    //Criacao de um conjunto de threads
    #pragma omp parallel shared(M_New, M_Old) private(i, j)
    {
        //Guarda a ultima solucao em M_Old
        #pragma omp for
        for(i = 0; i < N; i++) {
            for(j = 0; j < N; j++) {
                M_Old[i][j] = M_New[i][j];
            }
        }

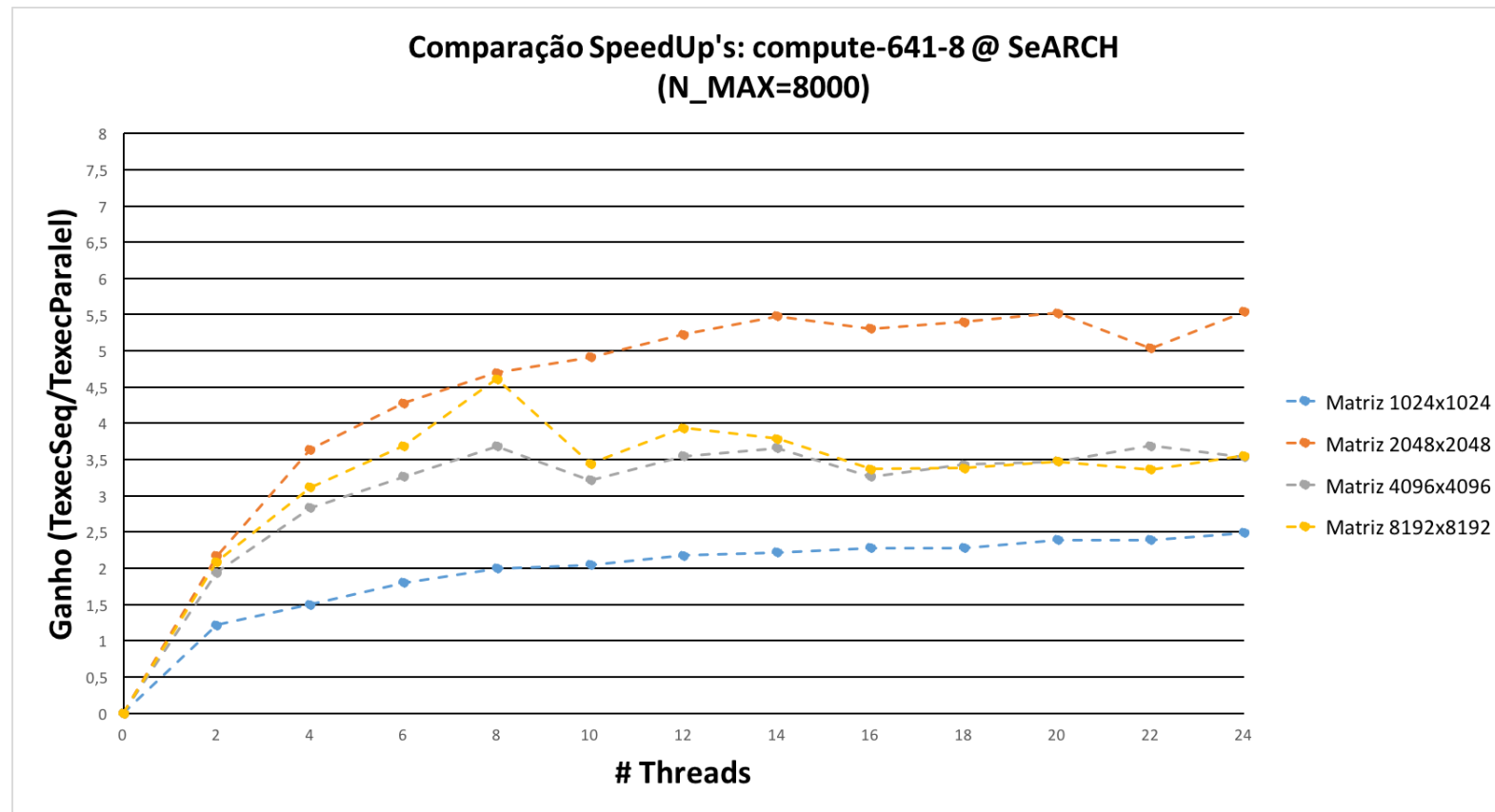
        //Calcula os novos valores dos pontos interiores para M_New.
        #pragma omp for
        for(i = 1; i < N-1; i++) {
            for(j = 1; j < N-1; j++) {
                M_New[i][j] = (M_Old[i-1][j] + M_Old[i+1][j] + M_Old[i][j-1] + M_Old[i][j+1] + M_Old[i][j])/5;
            }
        }
    }
}
```

Utilização das primitivas:

- #pragma omp parallel shared e private
- #pragma omp for

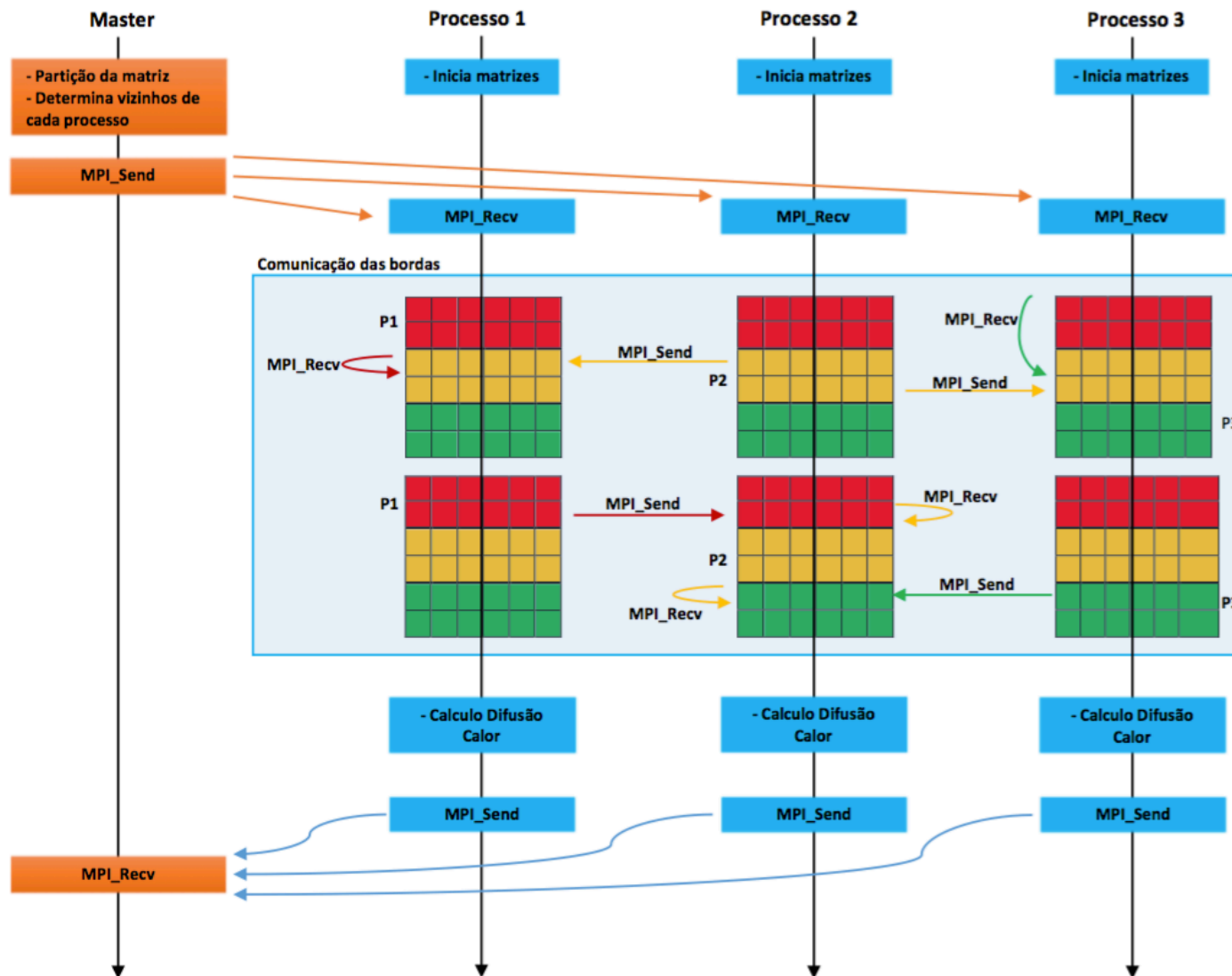


SpeedUp's – OpenMP Nodo 641



OpenMPI

Solução Paralela - OpenMPI



- Boa distribuição de carga:
 - Cada processo é responsável por um nº mínimo de linhas igual em todos os processos e no pior caso é processado esse nº mínimo + 1 linhas
- Processos vizinhos de outro processo são os processos imediatamente acima e abaixo do processo, em relação à zona da matriz que é responsável.
- Comunicação das linhas de borda entre processos vizinhos efetuada em duas fases consecutivas:
 - 1º: se processo PAR então ENVIA, se ÍMPAR então RECEBE
 - 2º: se processo ÍMPAR então ENVIA, se PAR então RECEBE

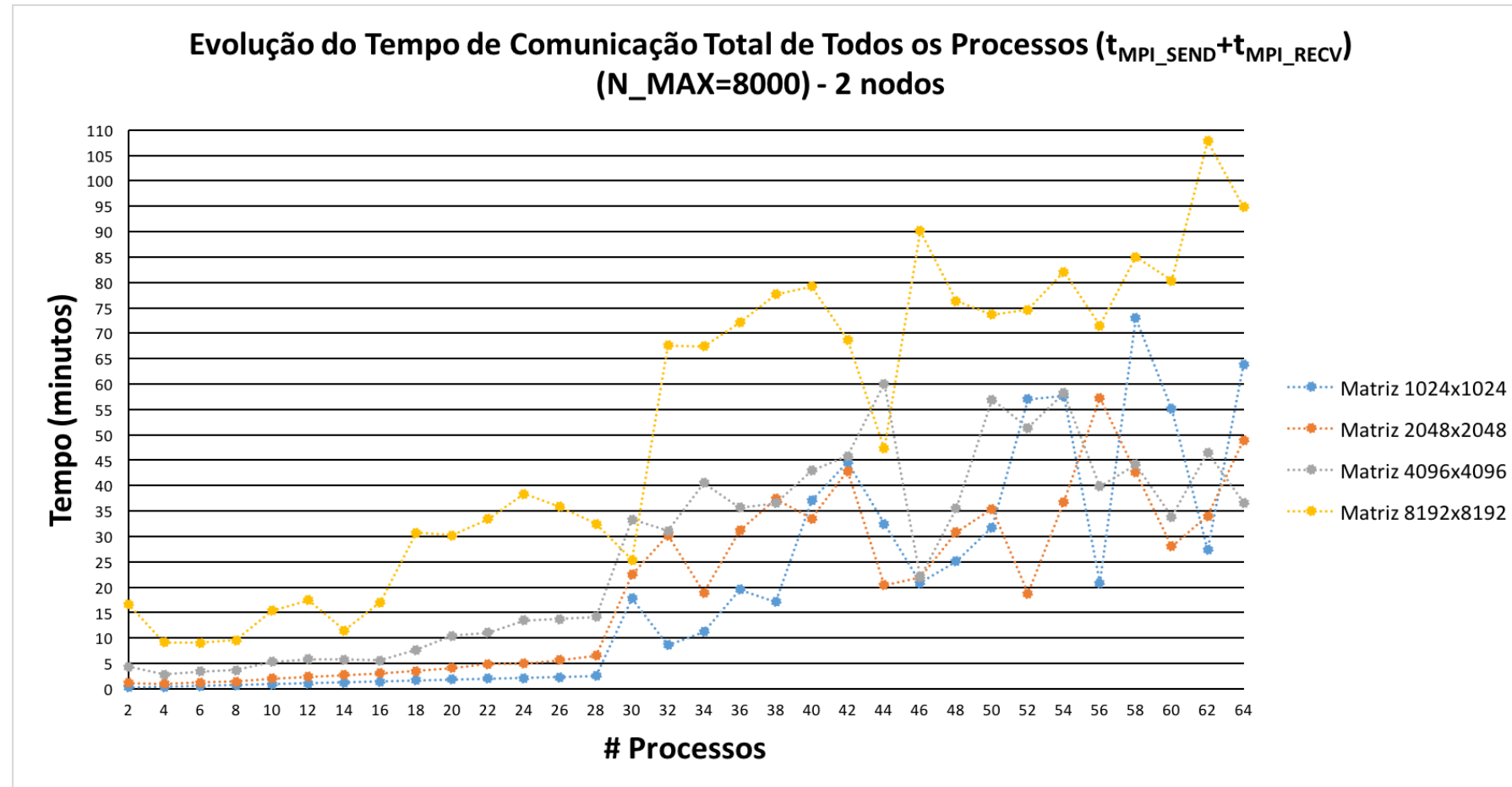


Testes Efetuados - OpenMPI

- Comunicação Ethernet
 - 1 nodo 641 até 32 processos
 - 2 nodos 641 até 64 processos
 - 4 nodos 641 até 128 processos (falta resultados)
- Comunicação Myrinet
 - 1 nodo 662 até 8 processos
 - 4 nodos 662 até 32 processos (falta resultados)
- Nº Máximo de Iterações = 8000

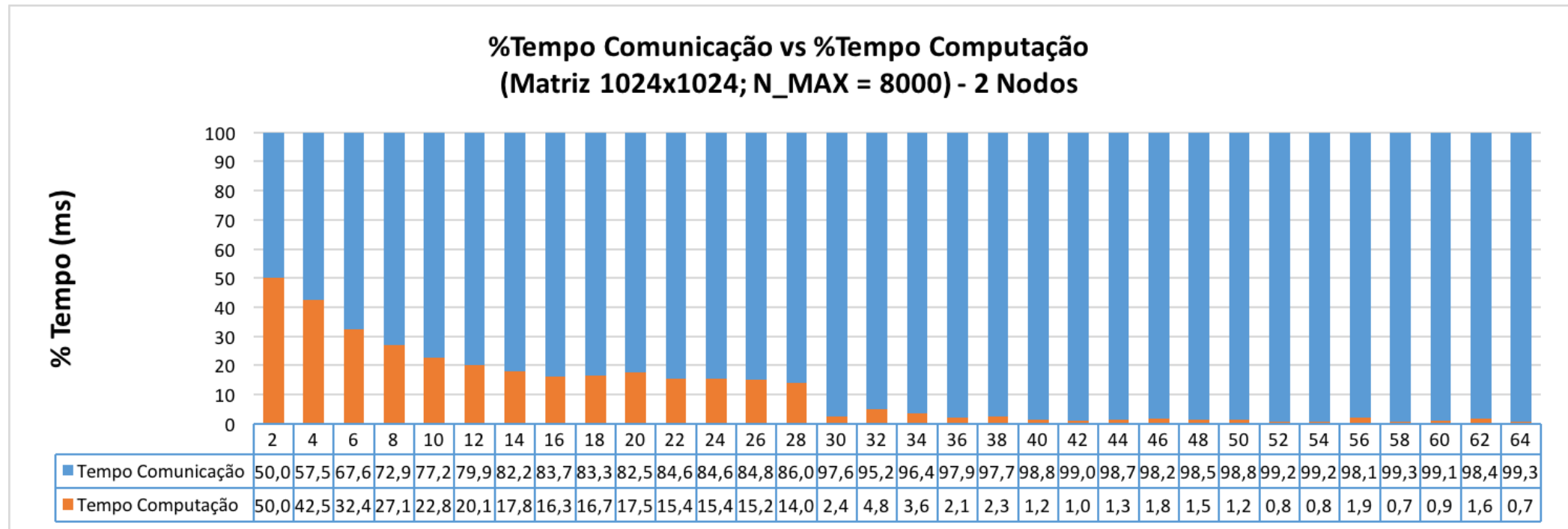


Evolução do Tempo de Comunicação – 2 nodos 641 por Ethernet



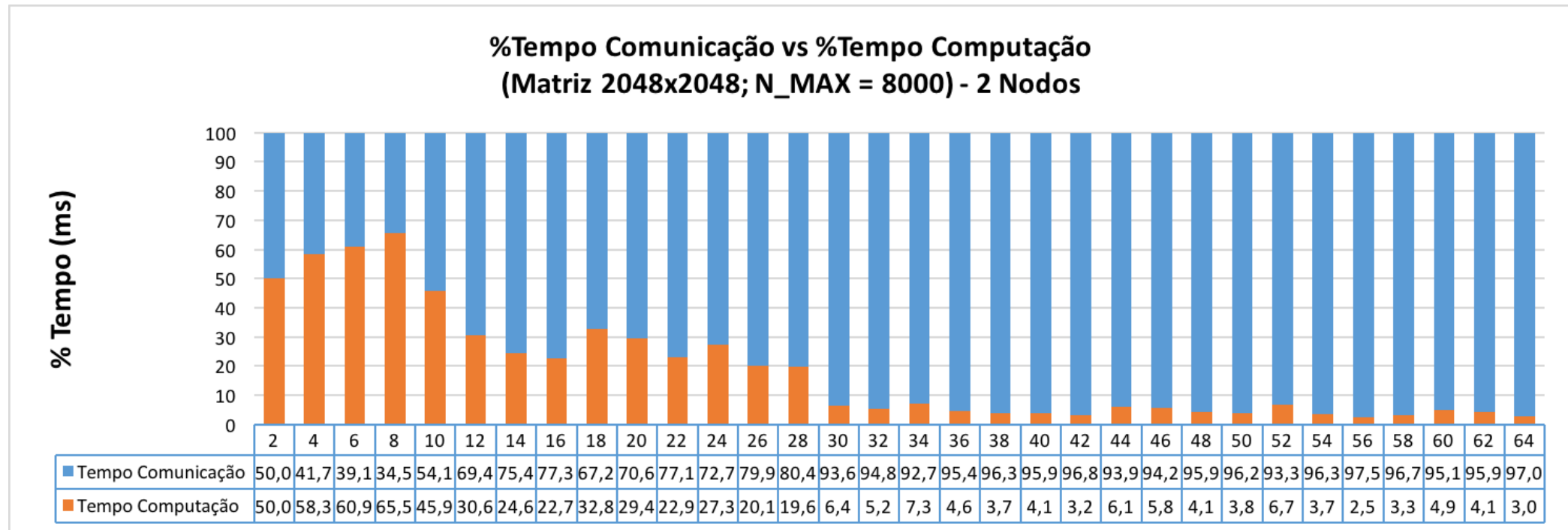
%Tempo Comunicação vs %Tempo Computação – 2 nodos 641 por Ethernet

Matriz 1024x1024



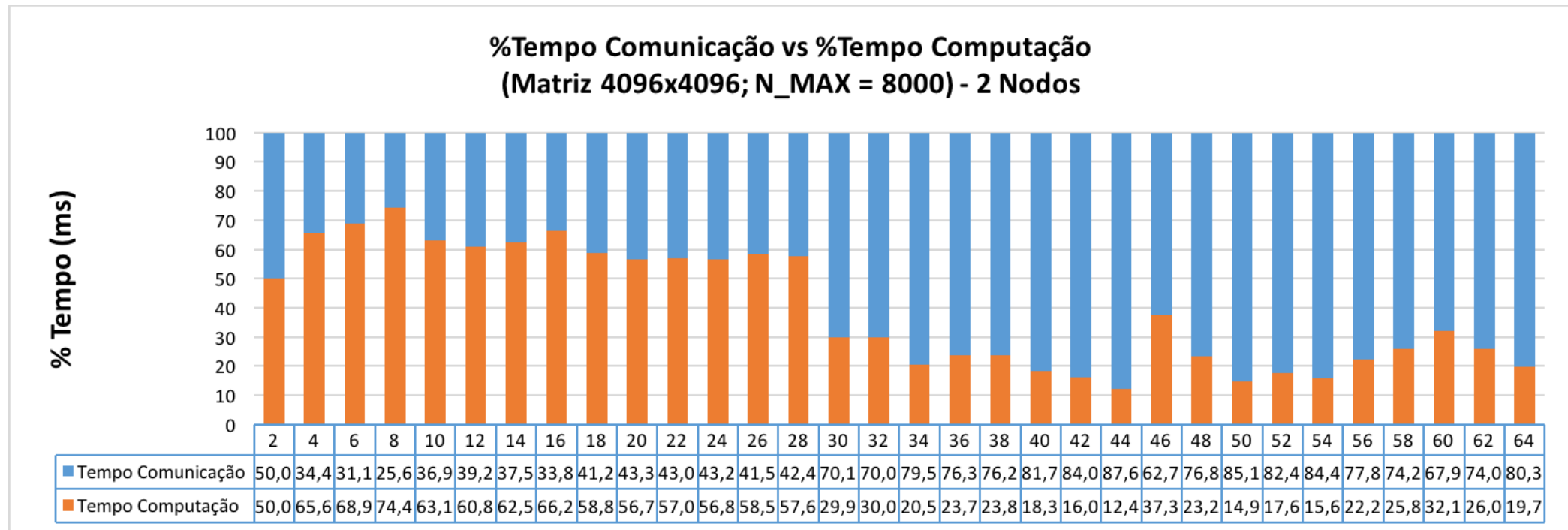
%Tempo Comunicação vs %Tempo Computação – 2 nodos 641 por Ethernet

Matriz 2048x2048



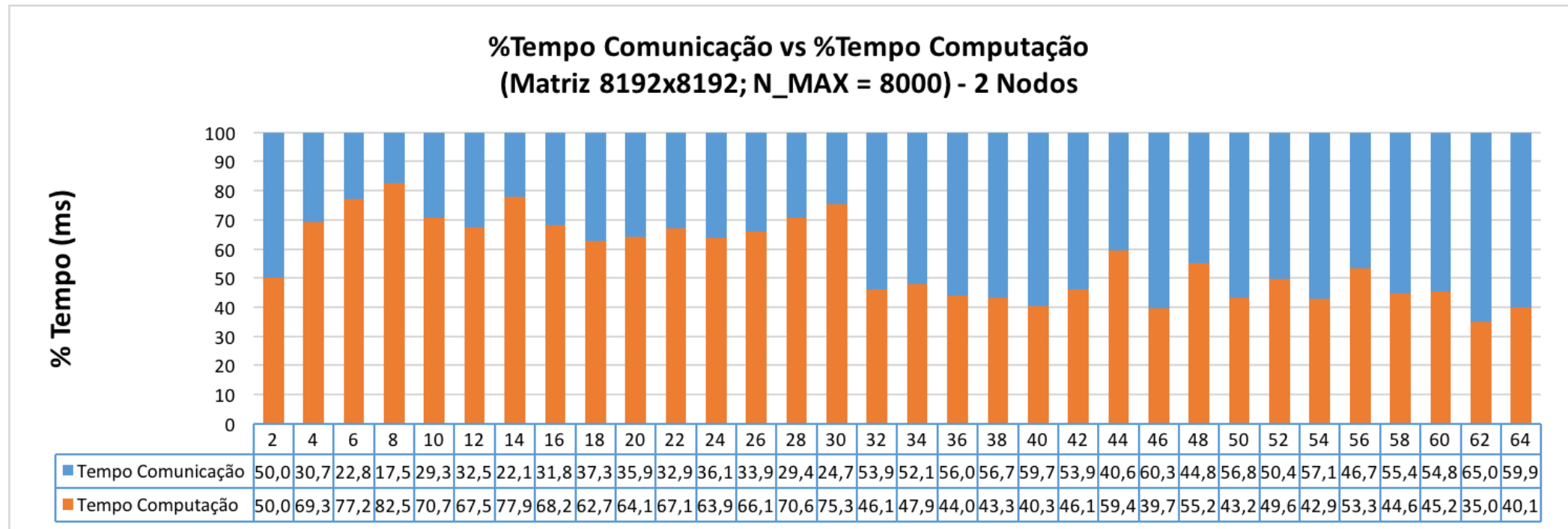
%Tempo Comunicação vs %Tempo Computação – 2 nodos 641 por Ethernet

Matriz 4096x4096

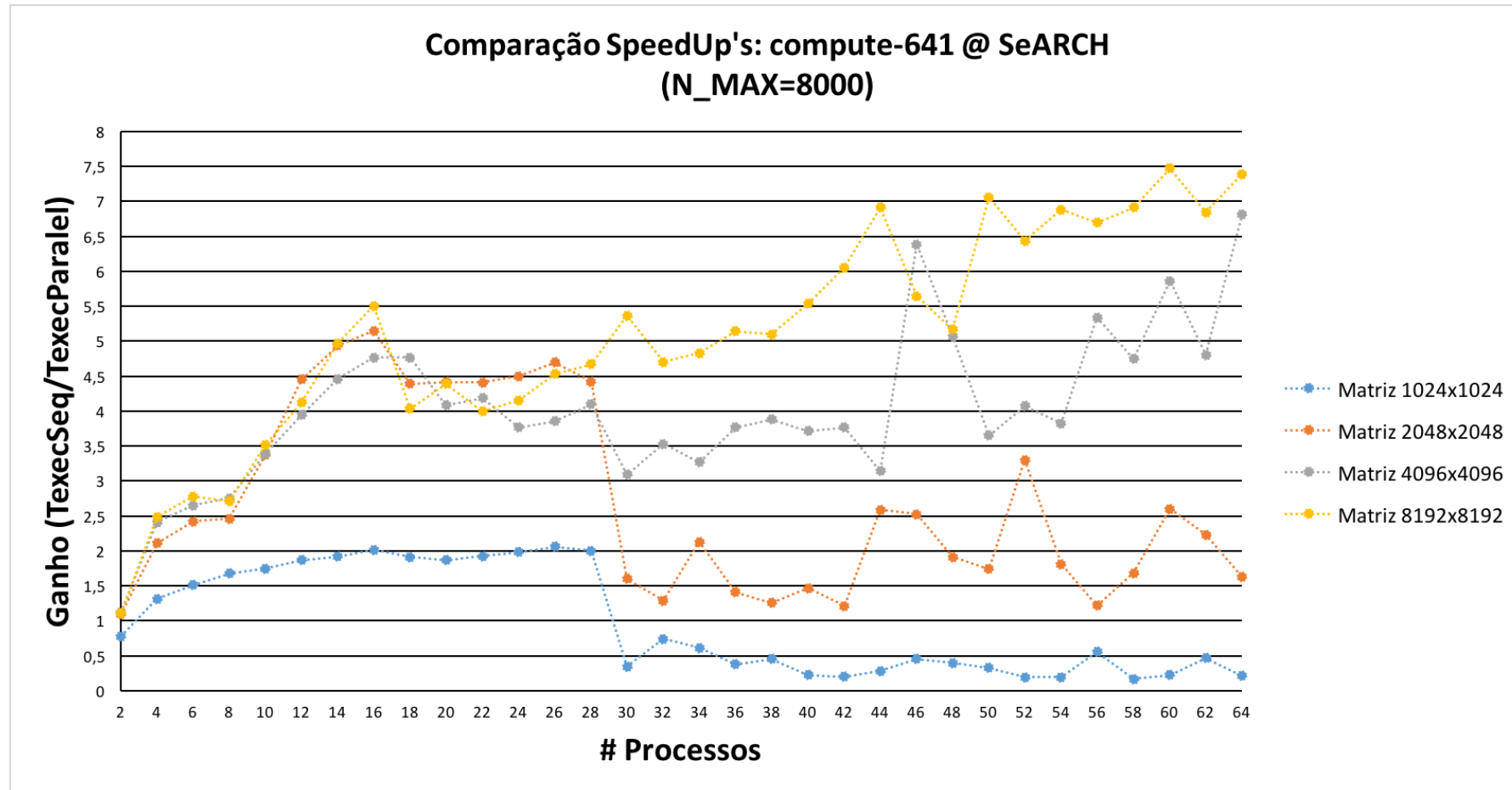


%Tempo Comunicação vs %Tempo Computação – 2 nodos 641 por Ethernet

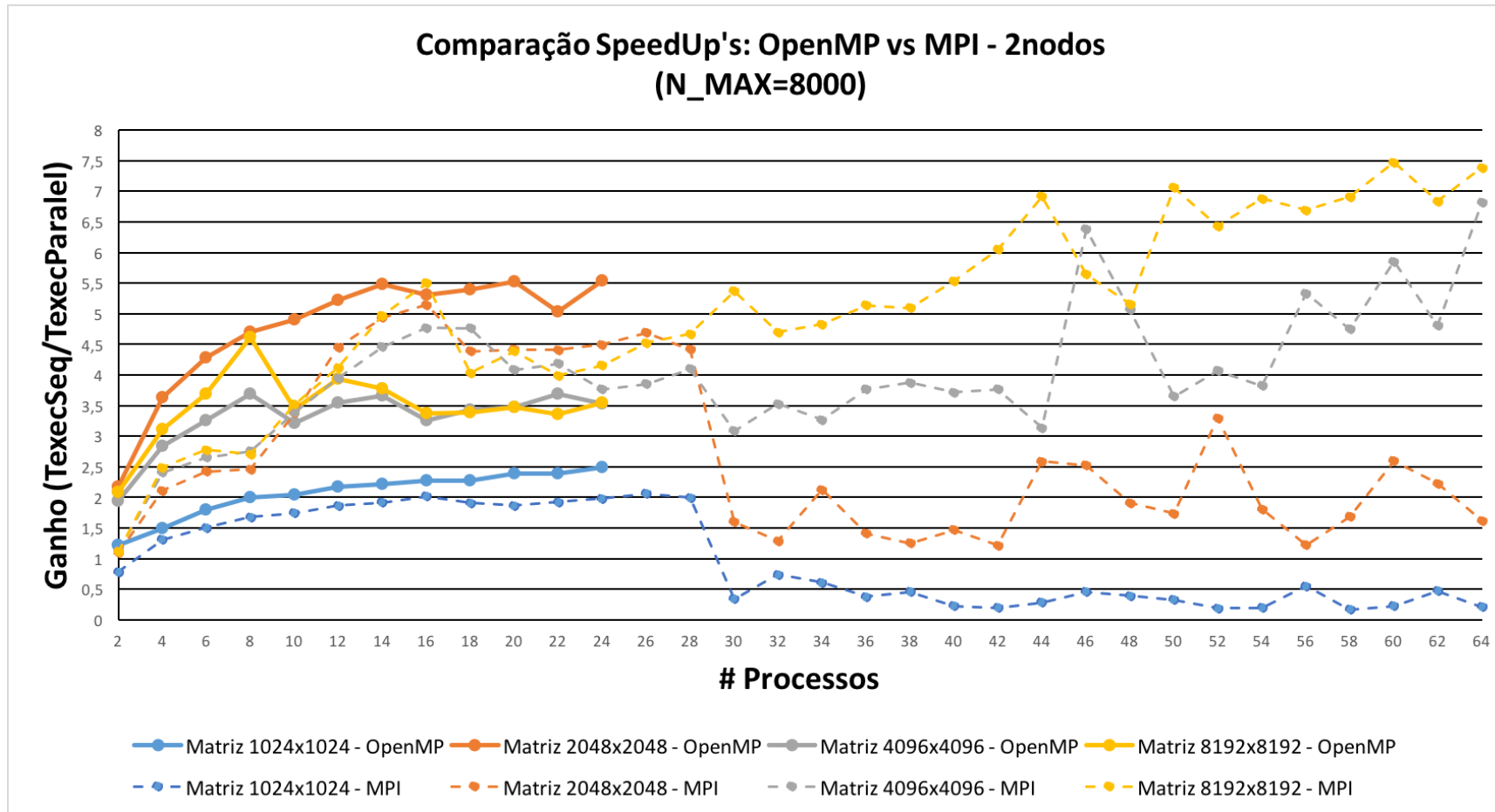
Matriz 8192x8192



SpeedUp's – 2 nodos 641 por Ethernet

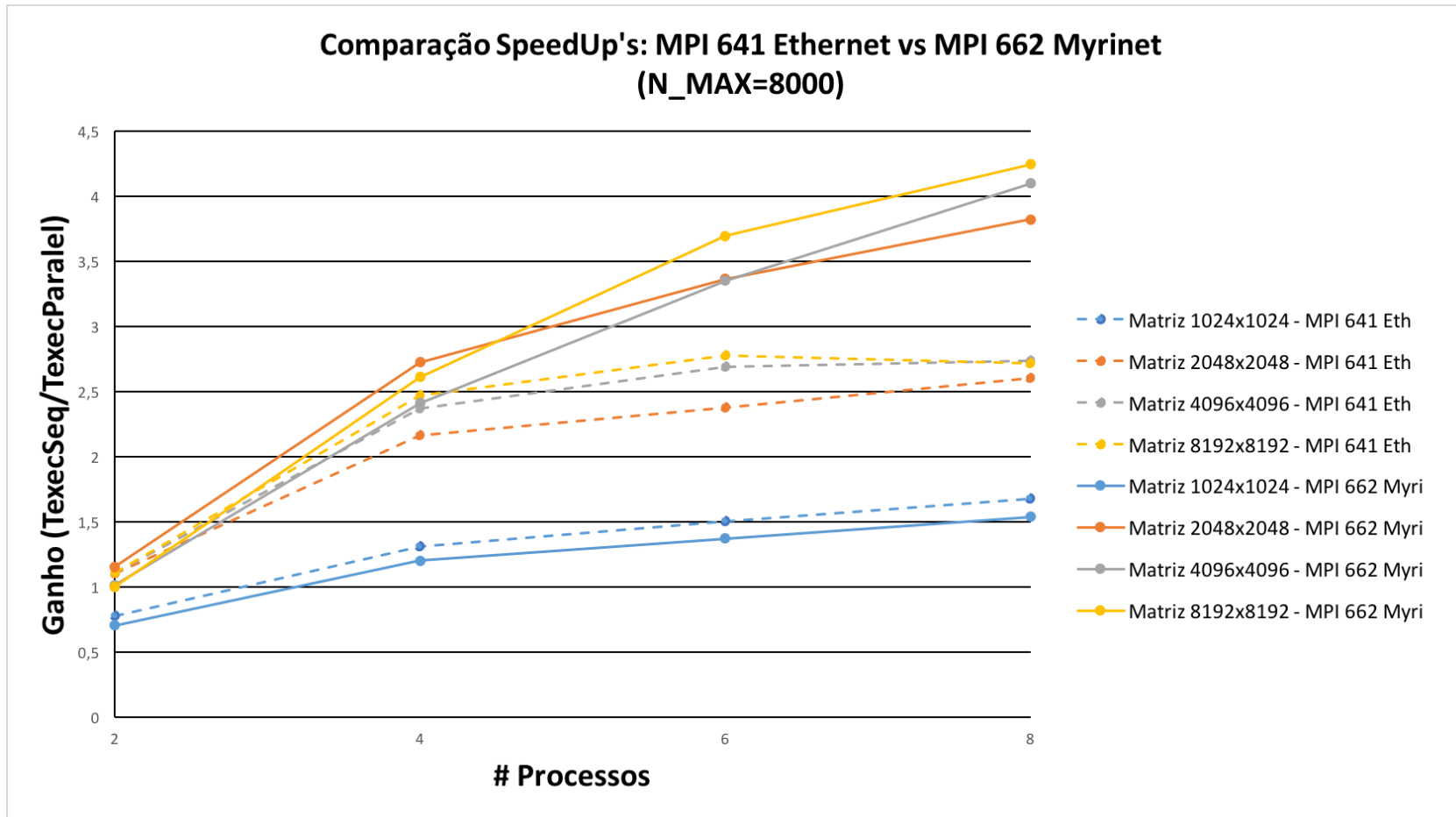


SpeedUp's OpenMP vs OpenMPI – nodos 641 por Ethernet



SpeedUp's MPI 641 Ethernet vs MPI 662 Myrinet

Myrinet – 8 Processos



Universidade do Minho
Ano letivo 2016/2017
Mestrado Integrado em Engenharia Informática

Paradigmas de Computação Paralela

Simulação do Processo de Difusão de Calor

Grupo 3
João Lopes a61077, Nuno Moreira a61017

Departamento de Informática
Escola de Engenharia

