

Modelos de Programação Paralela

ELC139 - Programação Paralela

João Vicente Ferreira Lima (UFSM)

Universidade Federal de Santa Maria

`jvlima@inf.ufsm.br`

`http://www.inf.ufsm.br/~jvlima`

2023/1

Outline

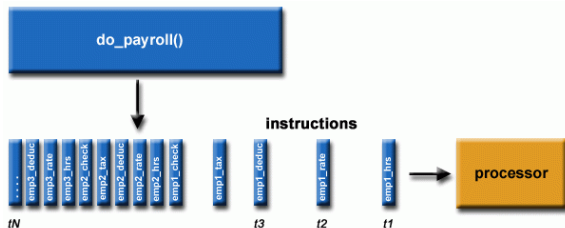
- 1 Programação Paralela
- 2 Metodologia PCAM
- 3 Conclusão PCAM

Outline

- 1 Programação Paralela
- 2 Metodologia PCAM
- 3 Conclusão PCAM

Programação Serial

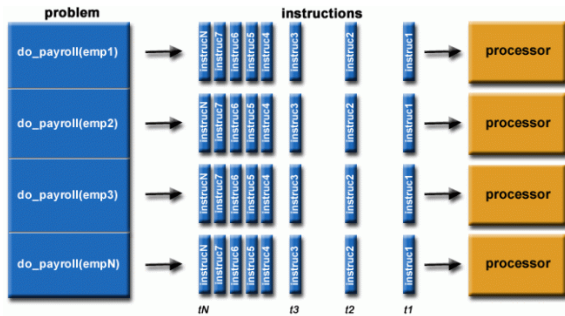
- Problema quebrado em uma série de instruções discretas
- Instruções são executadas sequencialmente uma depois da outra
- Execução em processador único



<https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial>

Programação Paralela

- O problema é quebrado em partes discretas que podem ser resolvidas concorrentemente
- Cada parte é quebrada em uma série de instruções
- As instruções podem executar simultaneamente em cada processador
- Um mecanismo “coordenador” deve ser empregado

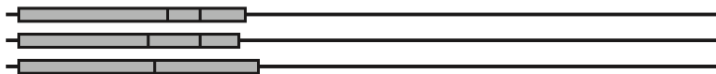


Concorrência vs Paralelismo

- **Concorrência** - múltiplas tarefas estão logicamente ativas ao mesmo tempo.
- **Paralelismo** - múltiplas tarefas estão realmente ativas ao mesmo tempo.



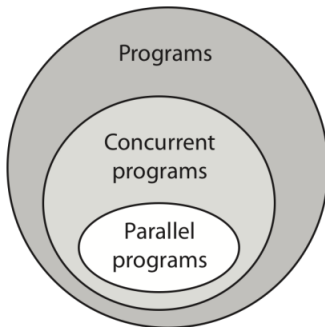
Concurrent, non-parallel execution



Concurrent, parallel execution

Concorrência vs Paralelismo

- **Concorrência** - múltiplas tarefas estão logicamente ativas ao mesmo tempo.
- **Paralelismo** - múltiplas tarefas estão realmente ativas ao mesmo tempo.



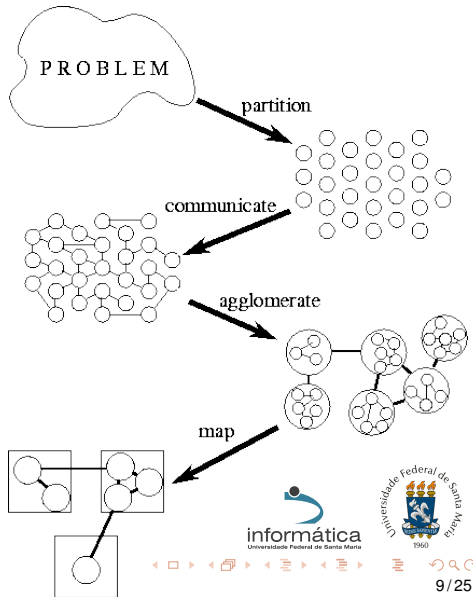
1 Programação Paralela

2 Metodologia PCAM

- Particionamento
- Comunicação
- Aglomeração
- Mapeamento

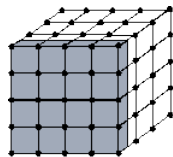
3 Conclusão PCAM

- Descrito por Ian Foster no livro *Designing and Building Parallel Programs*.
- Metodologia com quatro estágios.
 - Particionamento
 - Comunicação
 - Aglomeração
 - Mapeamento

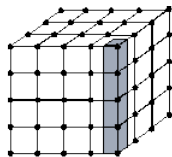


Particionamento

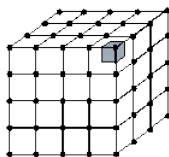
- Identificar o máximo de oportunidade de paralelismo
- Decompor o problema em dados e/ou operações
- Definimos a *granularidade*
 - Tamanho da tarefa gerada
 - Grão fino, ou grão grosso
- Checklist:
 - O número de tarefas é maior que o de processadores?
 - O tamanho das tarefas é
 - O número de tarefas escala com o tamanho?
 - É possível particionar de outras formas?



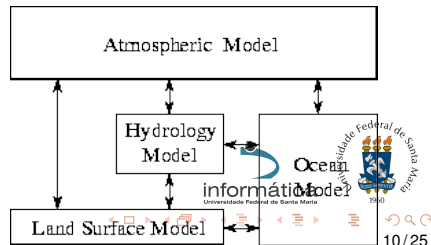
1-D



2-D

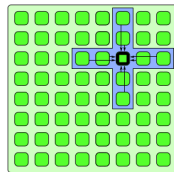
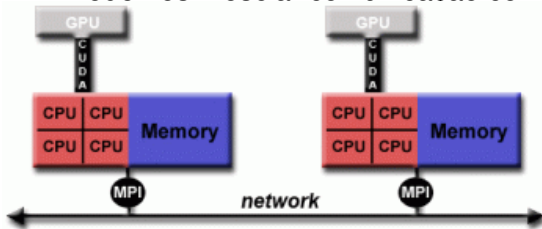


3-D



Comunicação

- O particionamento gera tarefas que executam concorrentemente mas não independentemente (em alguns casos)
- Comunicação deve satisfazer dependências
- Depende da arquitetura/plataforma
- Checklist:
 - As comunicações são uniformes?
 - A comunicação é para poucos vizinhos?
 - As comunicações podem ser concorrentes?
 - Podemos mesclar comunicação com processamento?

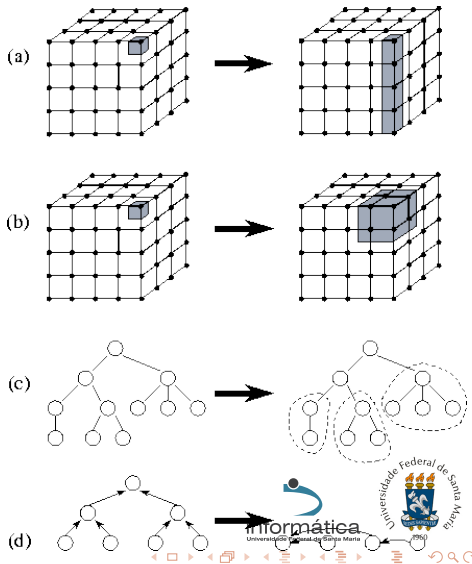


informatica
Universidade Federal de Santa Maria



Aglomeração

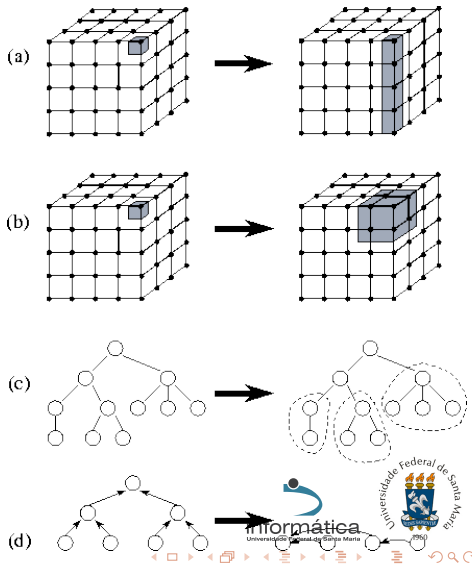
- As tarefas criadas são pequenas?
Eficientes?
- Na aglomeração, revisamos o
particionamento e comunicação
- Um grande número de tarefas pequenas
não produz um algoritmo paralelo eficiente
- Evitamos que o tempo de comunicação
domine a computação
 - Deve-se assumir que comunicações geram
sobrecusto considerável.
- Uma das técnicas é replicar/duplicar o
processamento para reduzir comunicação.
 - *Ghost zones*



Aglomeração

- Checklist:

- Aglomeração reduz comunicações e aumenta localidade?
- Replicar/duplicar tem custo benefício?
- Agrupar deixa o grão homogêneo?
- As tarefas escalam?
- Sobrecusto do código paralelo com relação ao sequencial?



-
- task 0
- task 1
- task 2
- task 4
- work
- wait
- time

Mapeamento

row-wise distribution

P_0
P_1
P_2
P_3
P_4
P_5
P_6
P_7

column-wise distribution

P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
-------	-------	-------	-------	-------	-------	-------	-------

Introduction to Parallel Computing, Grama et al, 2003.

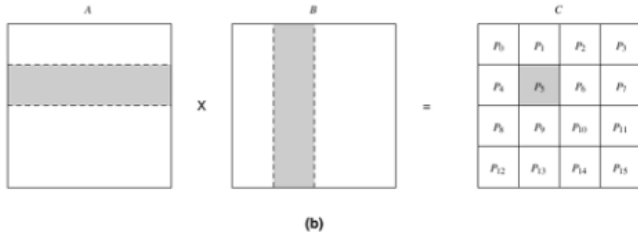
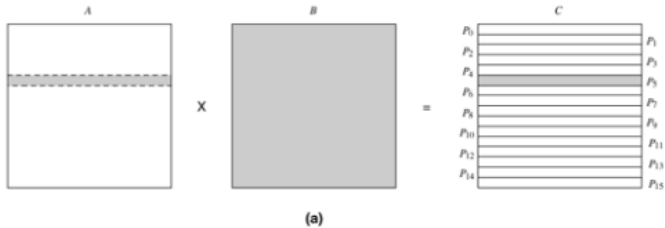
Mapeamento

P_0	P_1	P_2	P_3
P_4	P_5	P_6	P_7
P_8	P_9	P_{10}	P_{11}
P_{12}	P_{13}	P_{14}	P_{15}

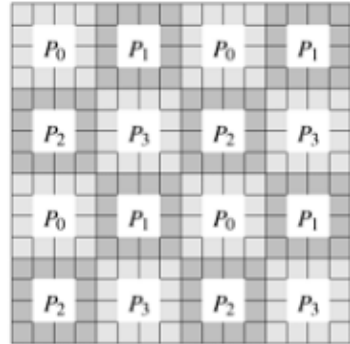
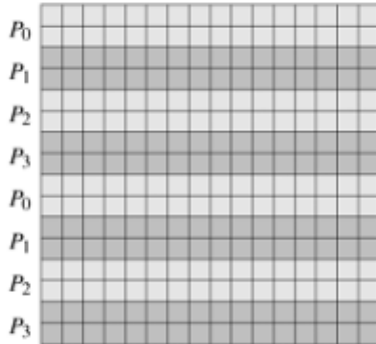
P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}

Introduction to Parallel Computing, Grama et al, 2003.

Mapeamento

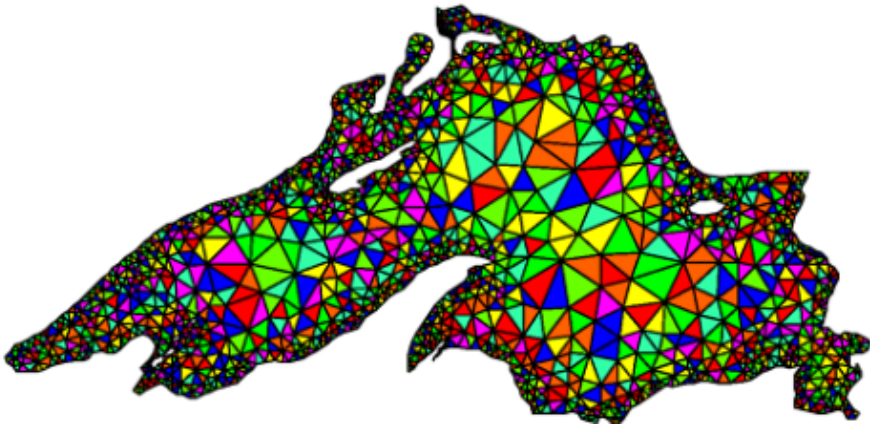


Mapeamento



Introduction to Parallel Computing, Grama et al, 2003.

Mapeamento



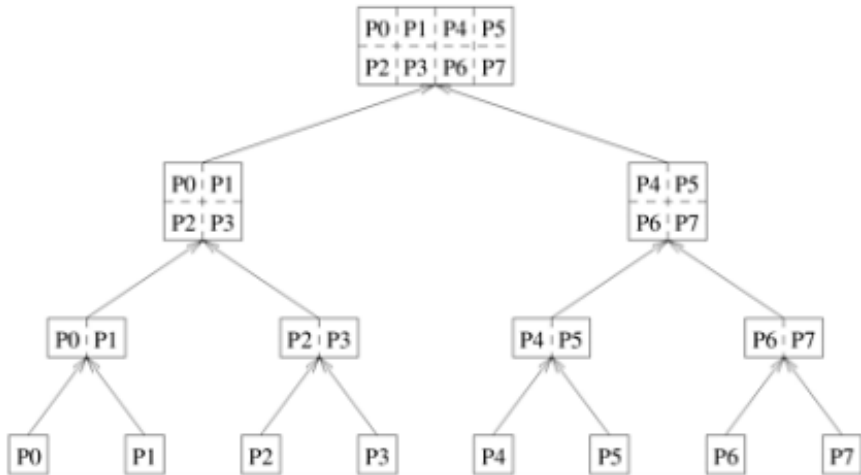
Introduction to Parallel Computing, Grama et al, 2003.

Mapeamento

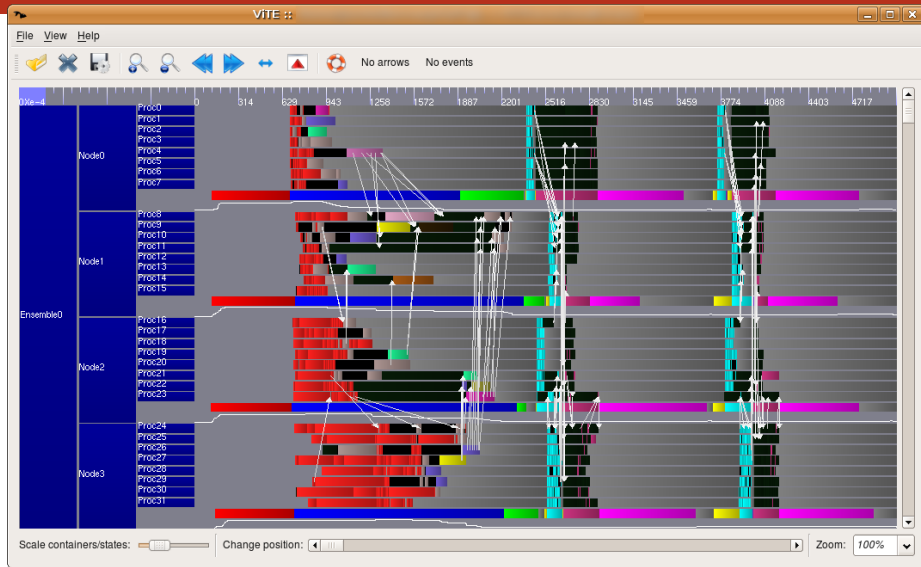


Introduction to Parallel Computing, Grama et al, 2003.

Mapeamento



Mapeamento

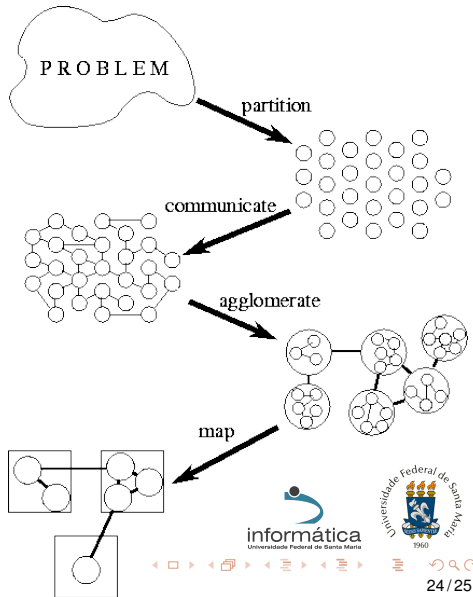


Outline

- 1 Programação Paralela
- 2 Metodologia PCAM
- 3 Conclusão PCAM

Conclusão PCAM

- Primeiro particionamos o problema em pequenos pedaços
 - Dados ou funcional
- Organizamos a comunicação entre tarefas quando necessário
- Analisamos a aglomeração para reduzir comunicação e sobrecusto paralelo
- Por fim, mapeamos tarefas para processadores.
 - Estático, balanceamento de carga, ou escalonamento



<https://joao-ufsm.github.io/par2023a/>

