

PCAM: Particionamento, Comunicação, Aglomeração e Mapeamento



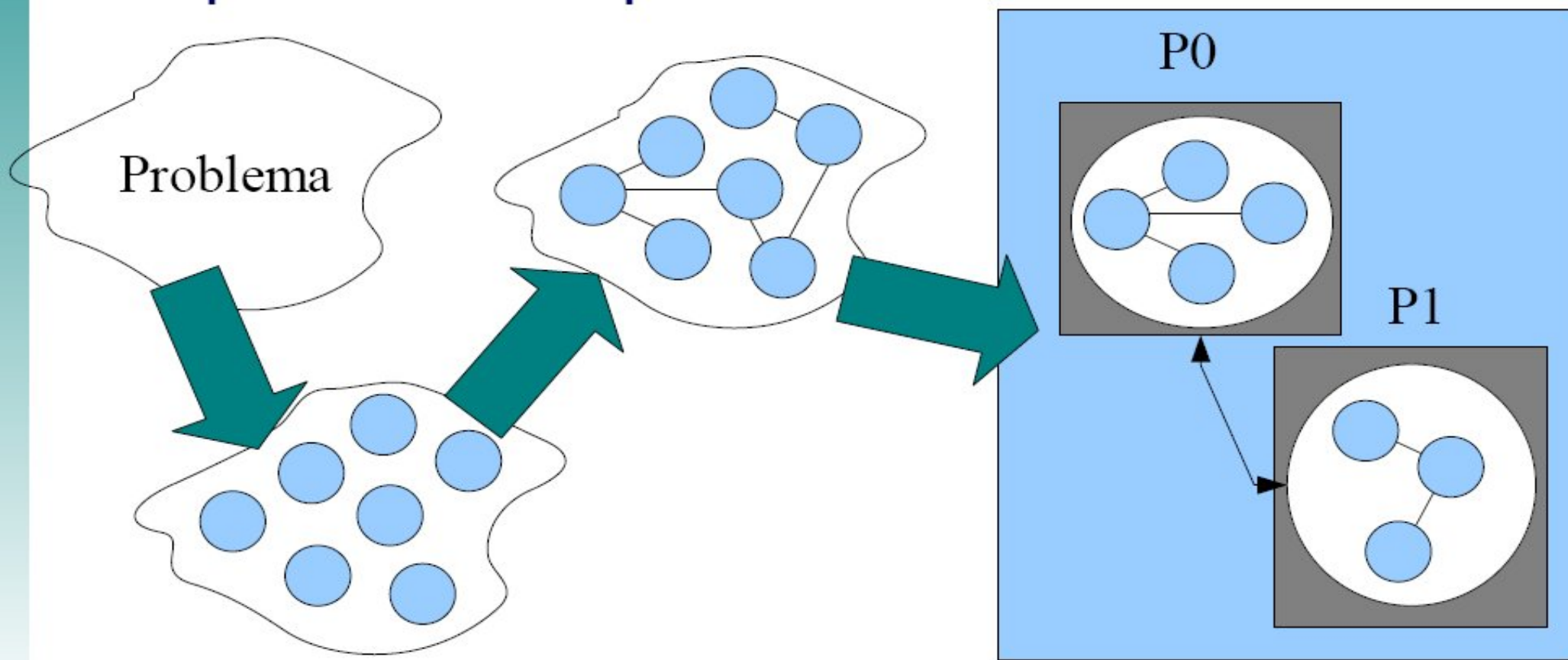
Programação Paralela Avançada - PPA

Mestrado em Computação Aplicação – MCA
Programa de Pós-Graduação em Computação Aplicada – PPGCA
Centro de Ciências Tecnológicas - CCT
Universidade do Estado de Santa Catarina – UDESC

Profs Maurício A. Pillon e Guilherme P. Koslovski

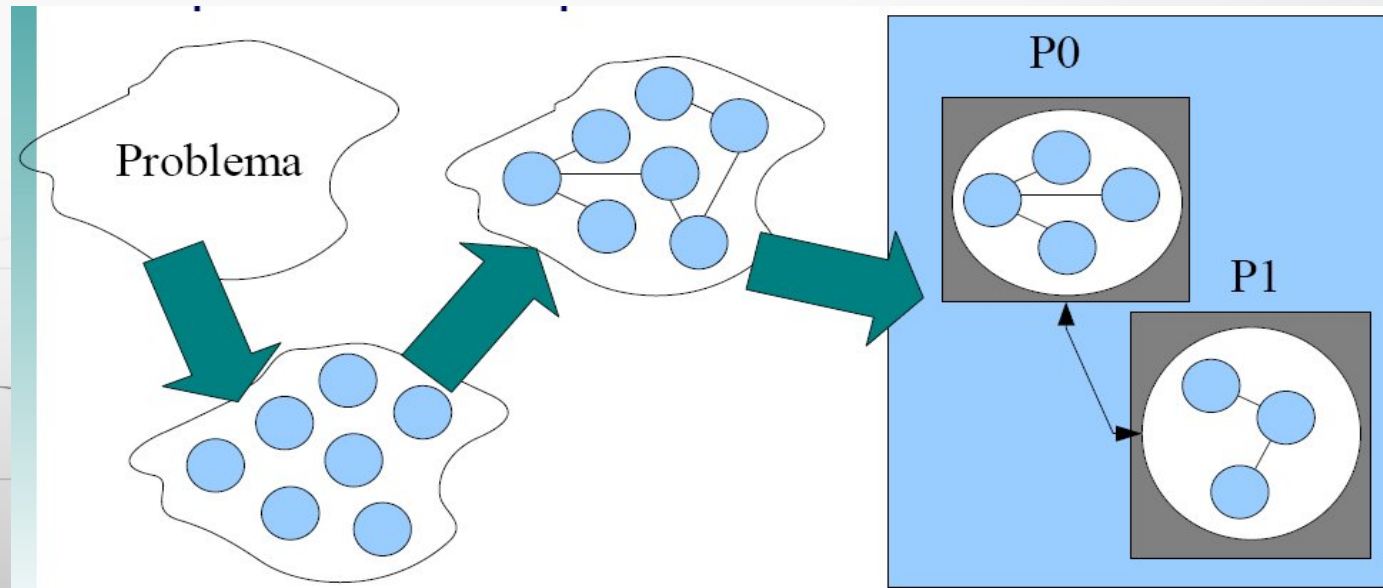
Linha de Sistemas Computacionais
Grupo de Pesquisa de Redes de Computadores e Sistemas Distribuídos
Laboratório de Pesquisa LabP2D

Revisando



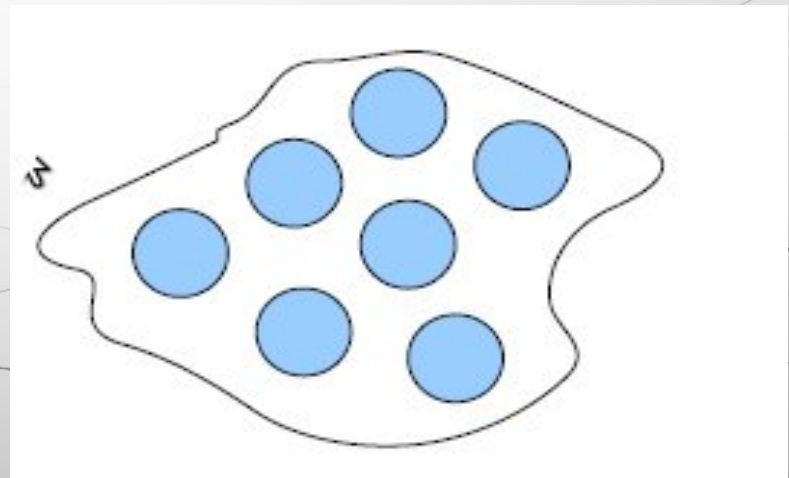
Metodologia de particionamento

- PCAM
 - Particionamento
 - Comunicação
 - Aglomeração
 - Mapeamento



PCAM: Particionamento

- Identificação de oportunidades para execução paralela
- Decomposição do problema
 - Subproblemas
- Decomposição?
 - Estrutural de dados
 - Estrutural de operações

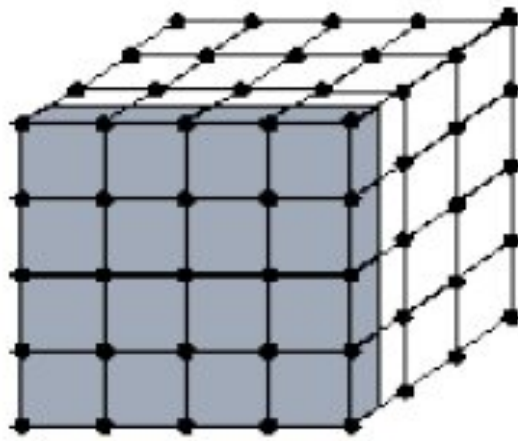


PCAM: Particionamento

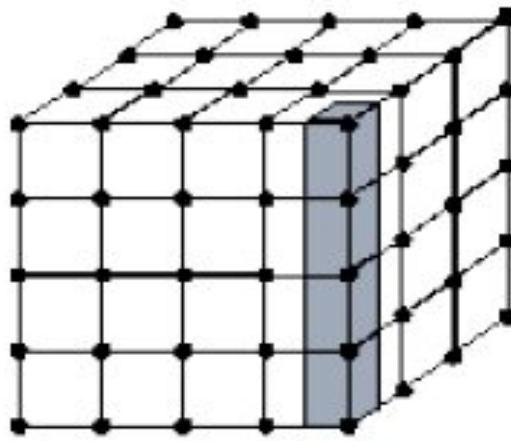
- Decomposição dos dados em pequenas partições de tamanhos semelhantes
- Decomposição das operações de acordo com o particionamento dos dados
- Tarefas resultantes: associação de partições de dados e operações associadas
- Comunicação entre tarefas pode ser necessária

PCAM: Particionamento

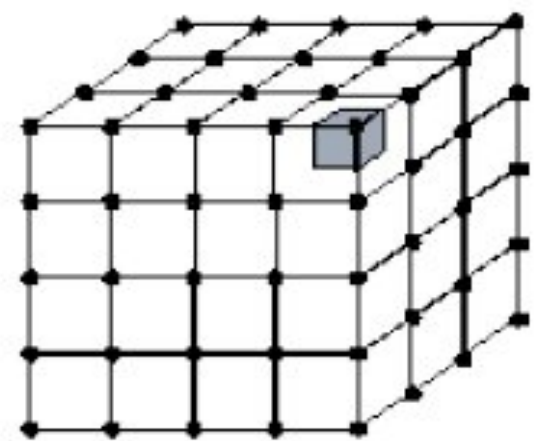
- Decomposição de dados
 - Podem existir diversas alternativas de particionamento
 - Ferramentas de auxílio (Metis, visualizadores de workflows: Moteur)



1-D



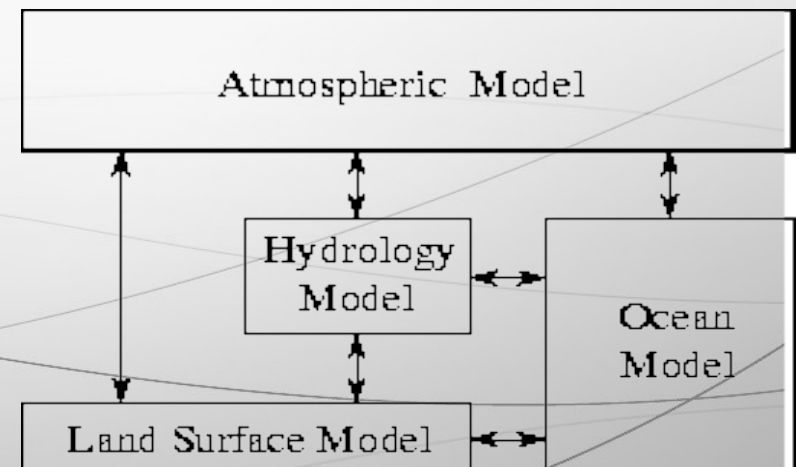
2-D



3-D

PCAM: Particionamento

- Decomposição funcional
 - Cada tarefa executa cálculos diferentes para resolver um problema
 - Tarefas podem ser executadas sobre os mesmos dados ou dados distintos
 - Exemplo: simulação de fenômenos atmosféricos



PCAM: Particionamento

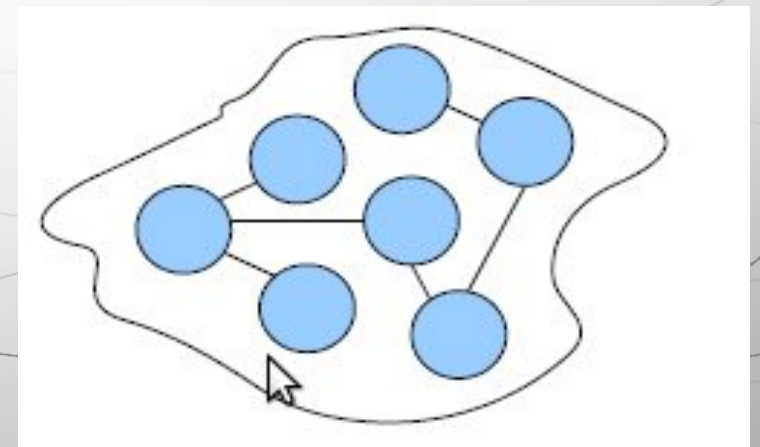
- Decomposição funcional
 - Decomposição dos dados de acordo com o particionamento das operações
 - Dados diferentes
 - Dados se sobrepõem
 - Replicação
 - Comunicação

PCAM: Particionamento

- Check list
 - O número de tarefas é maior que o número de processadores disponíveis?
 - As tarefas possuem aproximadamente o mesmo tamanho?
 - Você encontrou diversas opções de particionamento?

PCAM: Comunicação

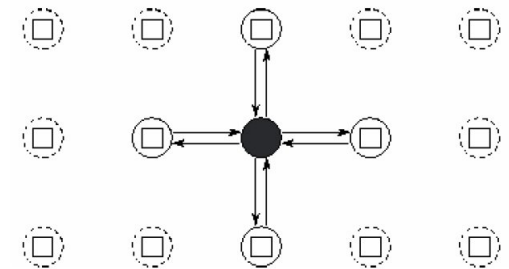
- Identificar e satisfazer as dependências e formas de cooperação entre tarefas
- A comunicação precisa ser minimizada
- Paradigmas: troca de mensagens e memória compartilhada



PCAM: Comunicação

- Local x Global
 - Local: Comunicação com um pequeno conjunto de “vizinhos”
 - Global: comunicação com muitas (ou todas) tarefas

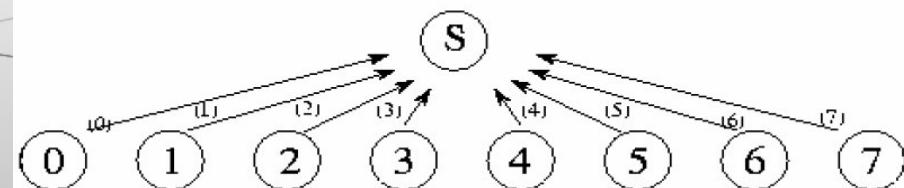
Local Communication



- Estruturada x Não-estruturada
 - Estruturada: árvores, grade
 - Não-estruturada: grafos arbitrários

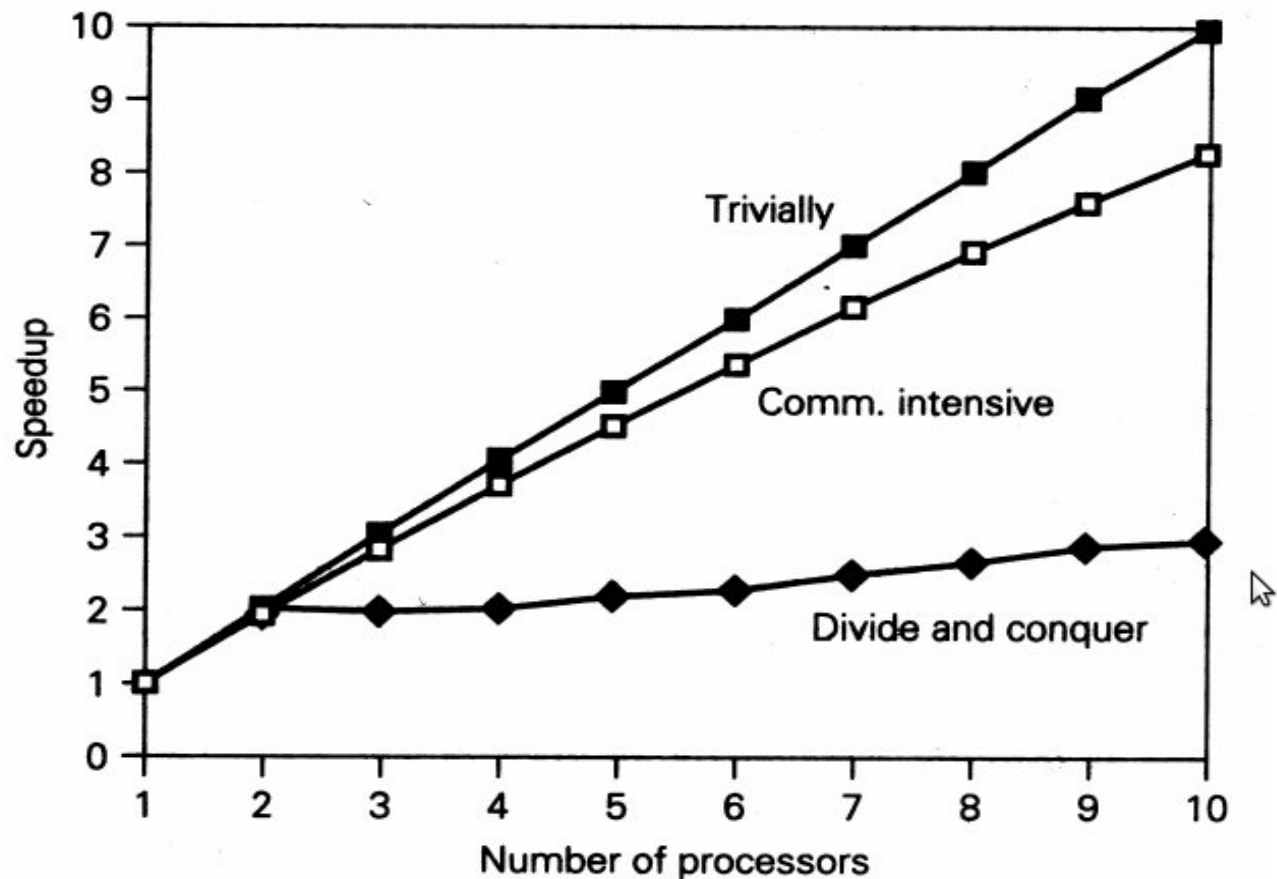
- Estática x Dinâmica
 - Estrutura/identidade dos pares muda ou não durante a execução

Global Communication



PCAM: Comunicação

Limites para o paralelismo

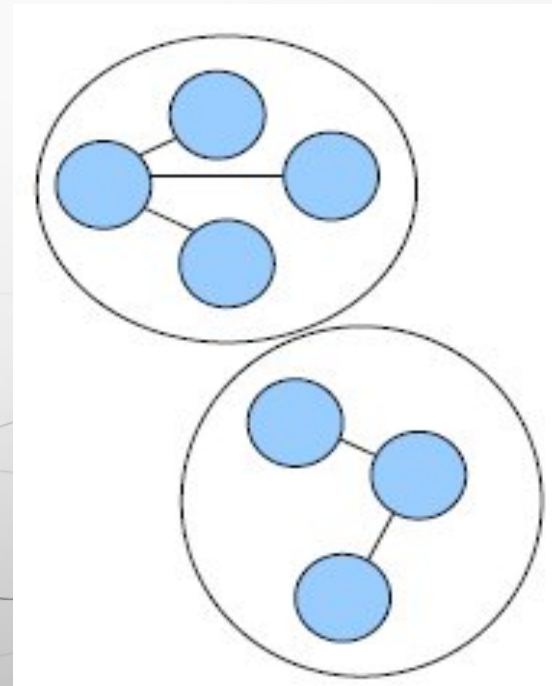


PCAM: Comunicação

- Checklist
 - Todas as tarefas realizam o mesmo número de comunicações?
 - As tarefas estão comunicando 'apenas' com seus vizinhos?
 - Processamento e comunicação podem ocorrer concorrentemente?

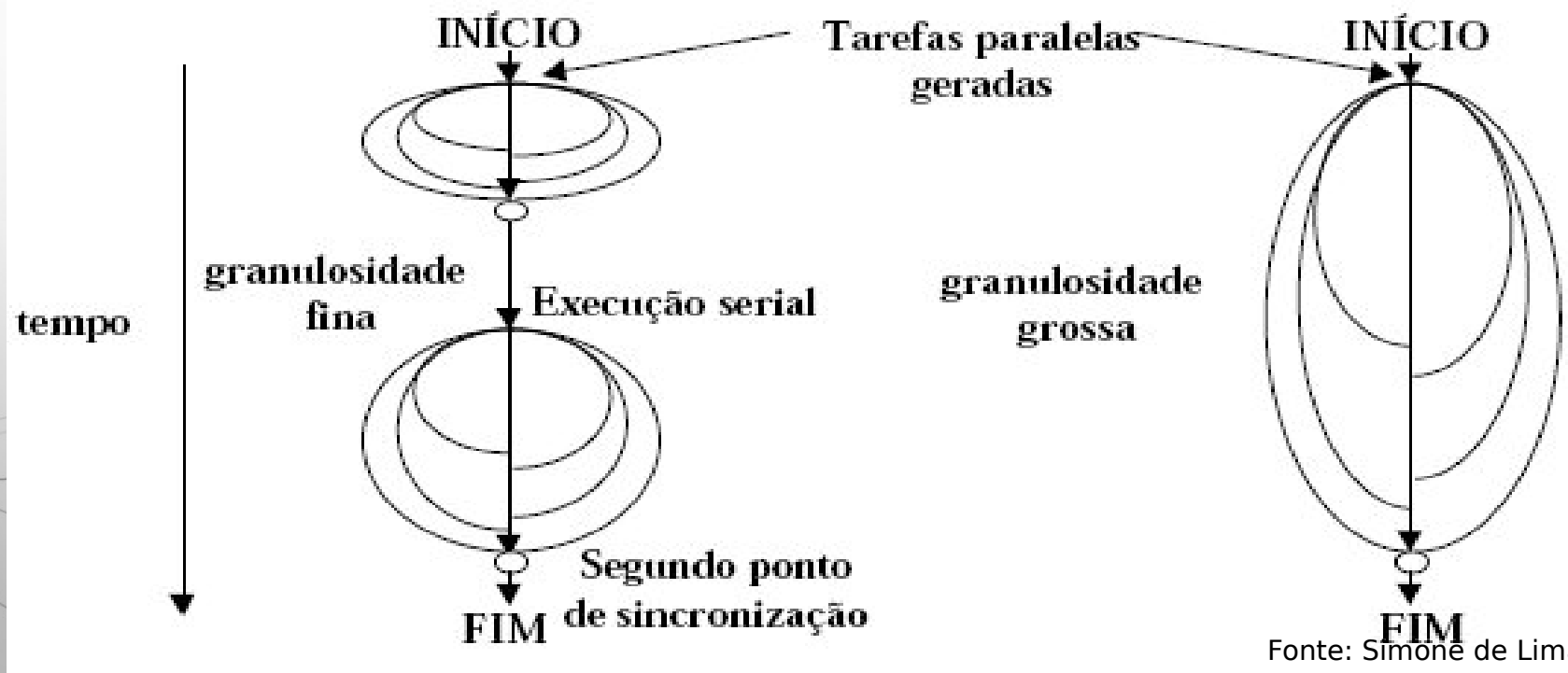
PCAM: Aglomeração

- Reduzir as comunicações (aumentar granulosidade), preservando o paralelismo
- Granulosidade: razão entre a quantidade de computação e a quantidade de comunicação
- Pode resultar em replicação de dados e/ou operações



PCAM: Aglomeração

- Granulosidade
 - Uma medida da razão entre a quantidade de computação realizada em uma tarefa e a quantidade de comunicação necessária



PCAM: Aglomeração

- Granulosidade
 - Nível de granulosidade varia de fina (muito pouco processamento por comunicação de byte) e grossa (muita computação por comunicação de byte)
 - Quanto mais fina a granulosidade menor a aceleração (sincronização!)

**Granulosidade
fina**

**Nível de
instrução**



**Loop com poucas
iterações**

**Granulosidade
grossa**

**Rotinas
longas**



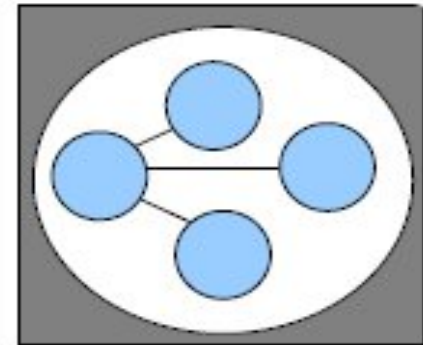
PCAM: Aglomeração

- Checklist
 - A aglomeração reduziu o número de comunicações?
 - Existem dados replicados? (é possível)
 - O número de tarefas resultantes afeta a escalabilidade do sistema?

PCAM: Mapeamento

- Alocação de tarefas aos processadores disponíveis
 - Tarefas independentes em processadores diferentes
 - Tarefas com dependências no mesmo processador
- Concorrência x localidade

P0



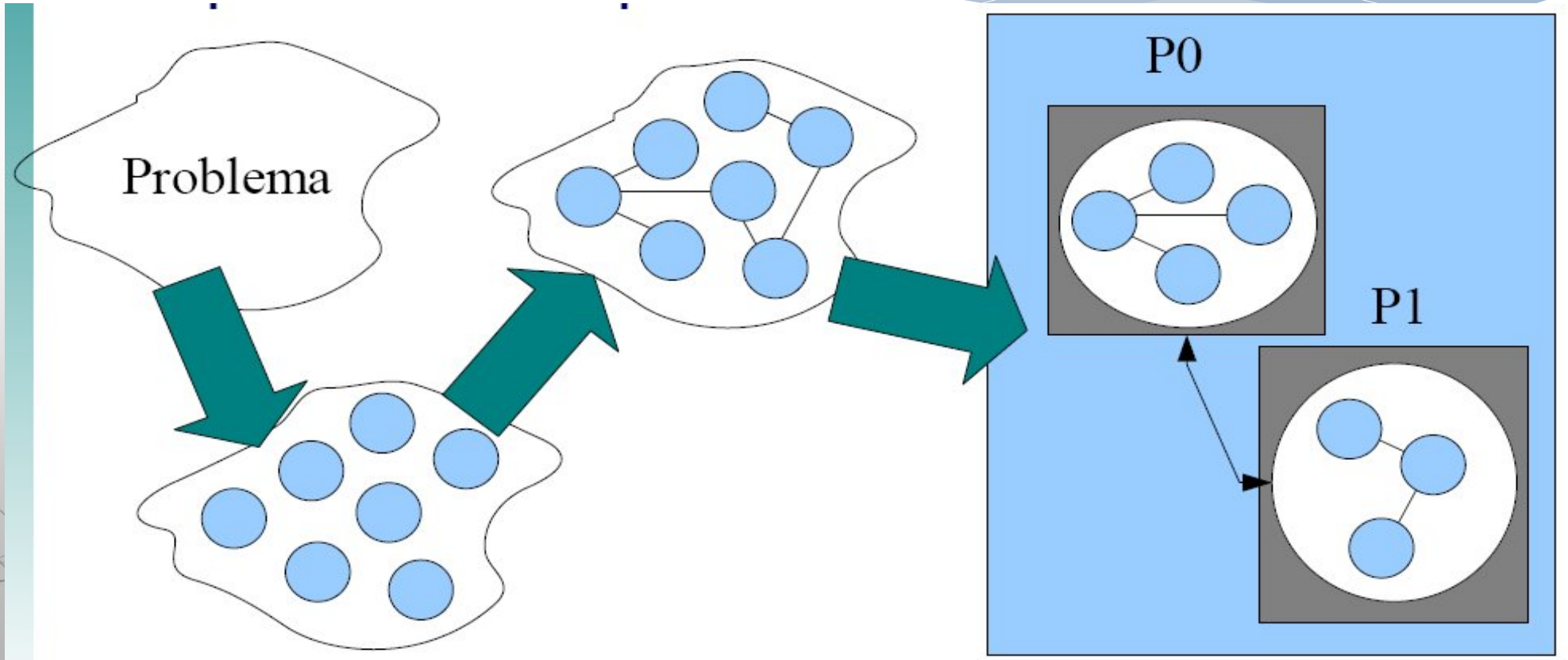
PCAM: Mapeamento

- Distribuição de carga
 - Mapeamento equitativo de tarefas considerando a capacidade do processador
- Mapeamento estático: definido no início da execução
- Mapeamento dinâmico: segue um balanceamento dinâmico

PCAM: Mapeamento

- Checklist
 - As tarefas comunicantes estão posicionadas no mesmo processador (ou próximas)?
 - Arquitetura física da rede
 - Heterogeneidade

PCAM



Agenda

- Projeto de programas paralelos
 - Metodologia de particionamento
- **Exemplo: Multiplicação de matrizes**
- Avaliação de desempenho de aplicações paralelas
- Considerações finais

Exemplo

- Multiplicação de matrizes

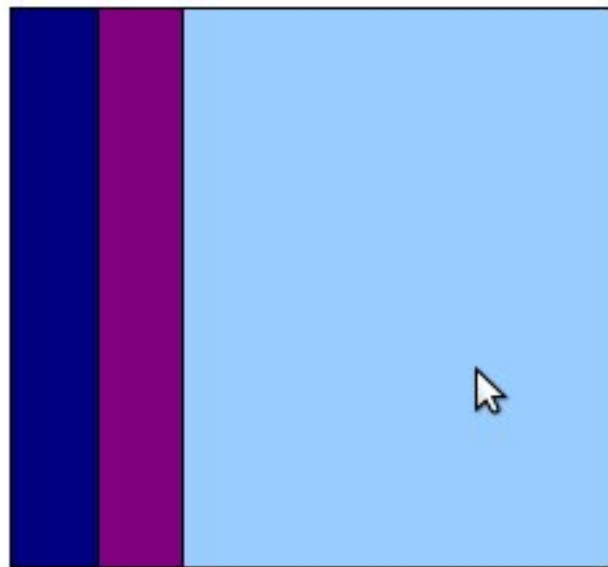
A

B

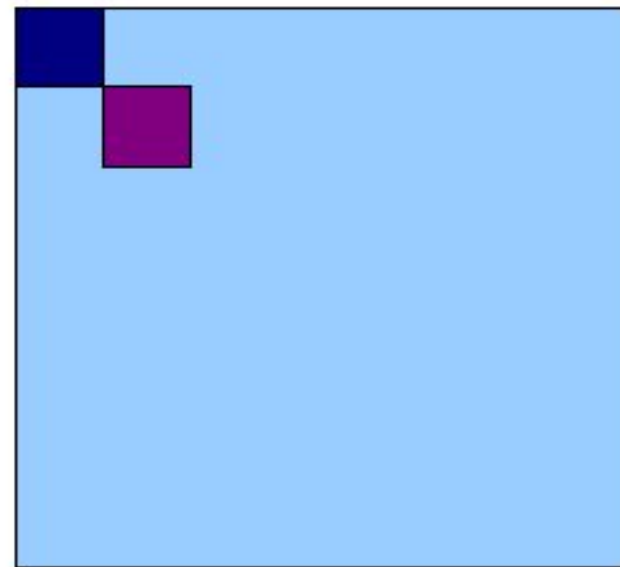
C



X



=



Exemplo

- Particionamento
 - Estrutural. Cada $c[i][j]$ pode ser calculado independentemente
- Comunicação
 - Tarefas que calculam diferentes $c[i][j]$ não cooperam
 - Para calcular $c[i][j]$ a linha $a[i][:]$ e a coluna $b[:,j]$ são necessárias
- Aglomeração
 - Exemplo: cada processador calcula algumas linhas da matriz C
 - Deve “conhecer” algumas linhas de A e de toda matriz B
- Mapeamento
 - Cada processador calcula o mesmo número de linhas
 - Distribuição estática

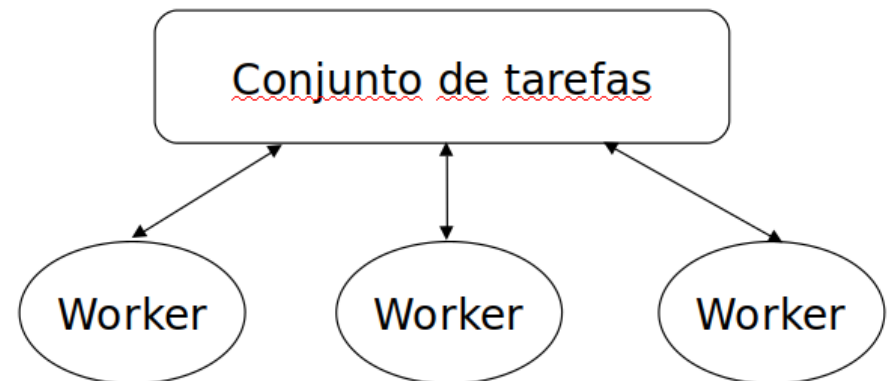
Modelos de aplicação

- As aplicações são modeladas usando um grafo que relaciona as tarefas e trocas de dados.
 - **Nós:** tarefas
 - **Arestas:** trocas de dados (comunicações e/ou sincronizações)
- Modelos básicos
 - Workpool
 - Mestre/escravo
 - Divisão e conquista
 - Pipeline
 - Fases paralelas

Modelos de aplicação

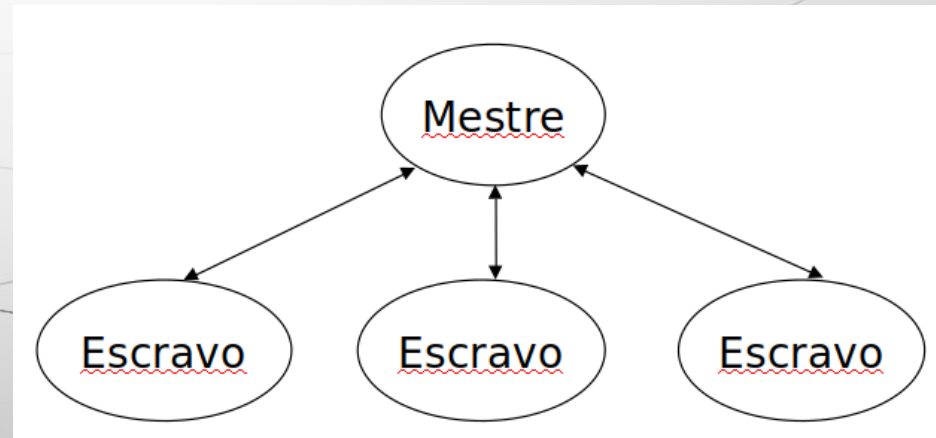
Workpool

- Tarefas disponibilizadas em uma estrutura de dados global (memória compartilhada)
- Sincronização no acesso à área compartilhada
- Balanceamento de carga



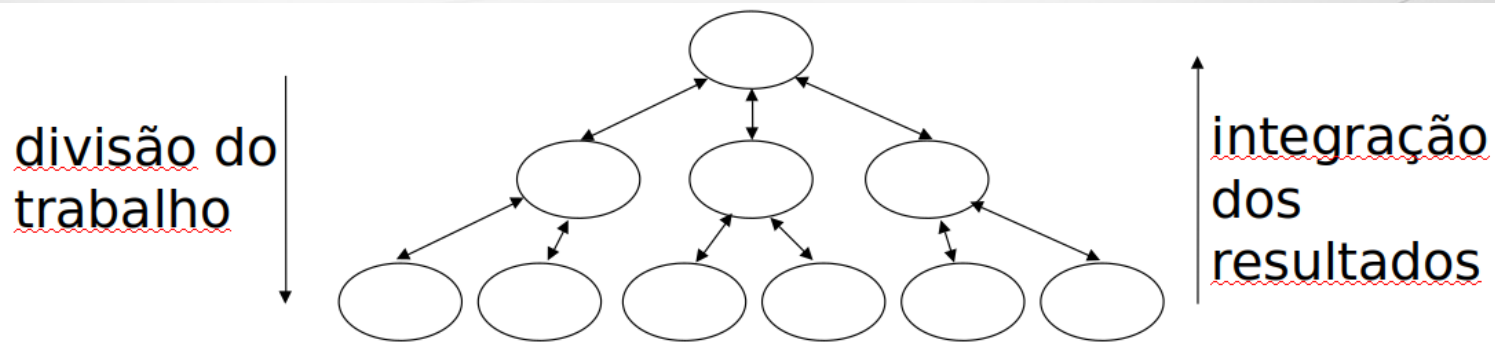
Modelos de aplicação

- Mestre / Escravo
 - Mestre escalona tarefas entre processos escravos
 - Escalonamento centralizado – gargalo



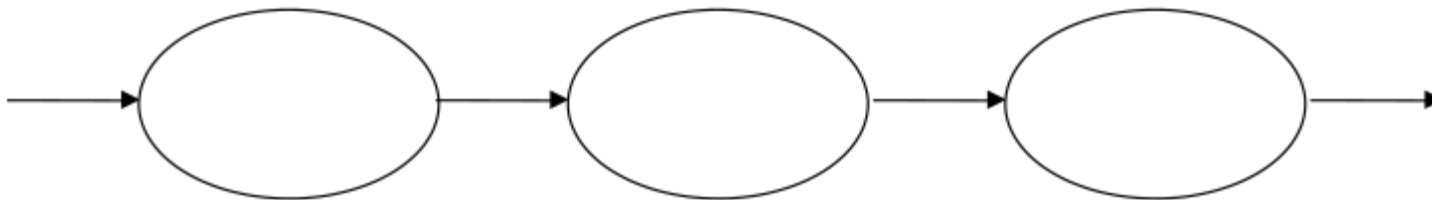
Modelos de aplicação

- Divisão e conquista (Divide and Conquer)
 - Processos organizados em uma hierarquia (pai e filhos)
 - Processo pai divide trabalho e repassa uma fração deste aos seus filhos
 - Integração dos resultados de forma recursiva
 - Distribuição do controle de execução das tarefas (processos pai)



Modelos de aplicação

- Pipeline
 - Pipeline virtual
 - Fluxo contínuo de dados
 - Sobreposição de comunicação e computação



Modelos de aplicação

- Fases paralelas
 - Etapas de computação e sincronização
 - Problema de balanceamento de carga
 - Processos que acabam antes
 - Overhead de comunicação

