Modelos para a Computação Paralela

Prof. Dr. Alfredo Goldman

www.ime.usp.br/~gold

gold@ime.usp.br

Motivação 1/2

- Existência de modelos bem aceitos para a computação sequencial:
  - Modelo de Von Neuman
    - Para a arquitetura
  - Máquina de Turing
    - complexidade
  - Com estes modelos fica fácil o desenvolvimento de programas...

Motivação 2/2

- Não existe um modelo universal para computação paralela;
- Existem duas características antagônicas:
  - simplicidade
  - realidade
    - Modelos simples não aproximam o comportamento de máquinas reais;
    - Modelo realistas são de difícil uso.

Taxonomia: uma proposta

- Modelos para a aplicação:
  - Representam o paralelismo de um algoritmo.
- Modelos de máquina:
  - Descrevem as características das máquinas.
- Modelos de execução:
  - Detalham a forma de programação

Criando uma aplicação paralela

1) Escolha de um modelo para a máquina;

2) Escolha de uma representação para o algoritmo;

3) Escolha de um modelo de execução;

Escalonamento

- Consiste em atribuir a cada tarefa, uma máquina e um tempo de início.
- Exemplos:
  - Processador e os processos;
  - Campeonato de Futebol;
  - Fotografias por satélite (Roadef'03);
  - O problema é geralmente difícil:
    - Duas máquinas e tarefas independentes.
- Mas, este não é o foco aqui....

Modelos para aplicação

Grafo de precedência 1/3

- Um grafo é a forma mais comum:
  - Grafos de fluxo de dados:
    - Os arcos representam a evolução dos dados
    - Pode-se transformar em grafo de precedência
  - Grafo de precedência;
  - Grafo de dependência.

Grafo de precedência 2/3

- O algoritmo é dividido em tarefas:
  - os vértices representam as tarefas;
  - os arcos as relações de dependência.

- Quando o grafo é formado em tempo de execução:
  - Escalonamento dinâmico;
- Veremos apenas exemplos de escalonamento estático

Grafo de precedência 3/3

- Termos importantes:
  - Sucessores;
  - Sucessores diretos;
  - Predecessores;
  - Predecessores Diretos;
  - Largura;
  - Caminho crítico;
  - Granularidade.

Como encontrar o grafo?

- Exemplo de multiplicação de matrizes:
  - Tradicional
  - Strassen

- Exercícios:
  - Calcular a soma de n números S_n
  - Imprimir todas as combinações $\binom{n}{n}$

Modelos para máquinas

Primeira proposta

- Feita por Flynn em 1966:
  - Classifica as máquinas conforme os fluxos:
    - de instruções
    - de dados

Dados / instr.	Simples	Múltiplo
Simples	SISD sequêncial	SIMD
Múltiplo	MISD	MIMD

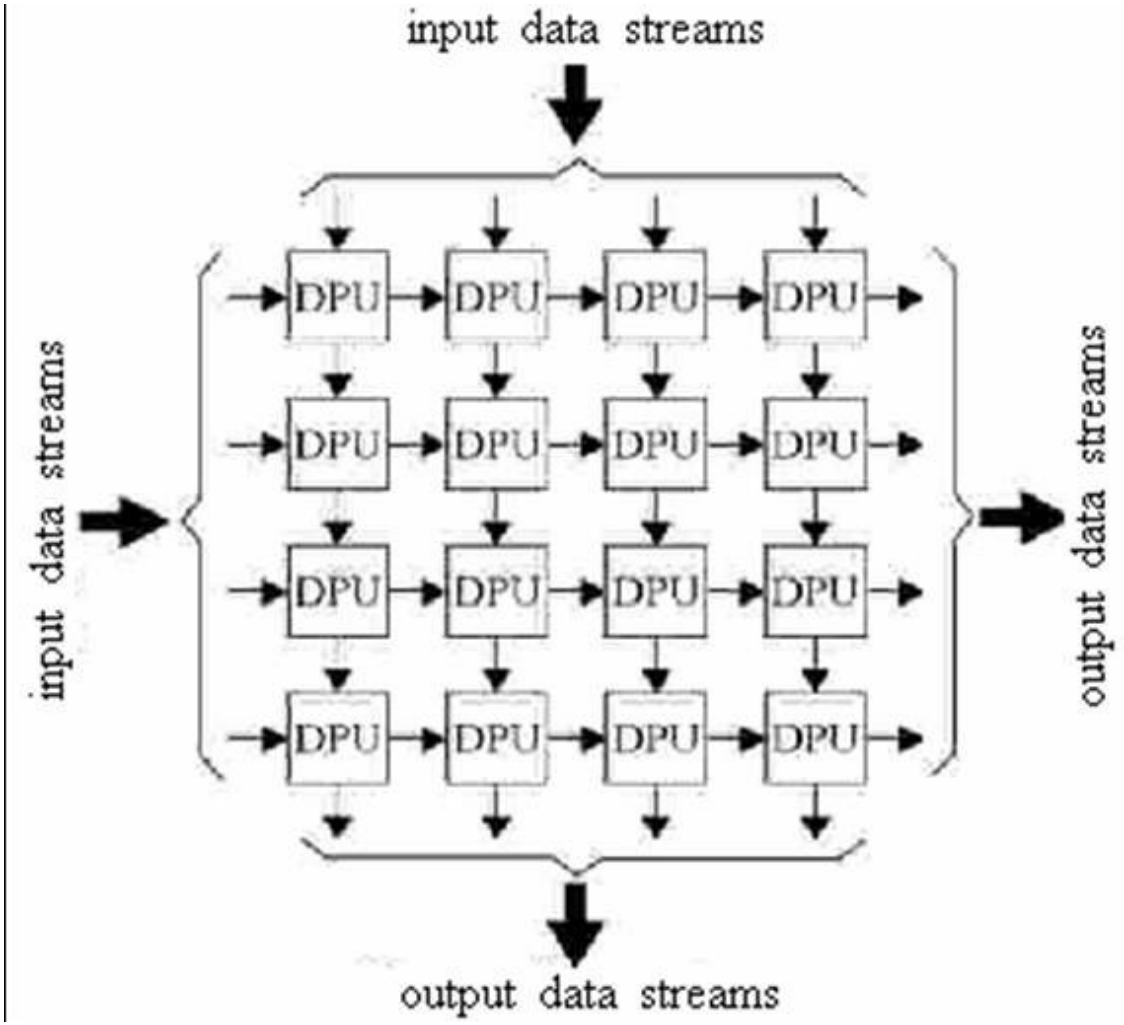
Outras classificações

- Velocidade de comunicação
  - Multi-computadores
  - Multi-processadores
- Tipo de acesso a memória
  - UMA - uniforme
  - NUMA - memória local
- Máquinas vetoriais

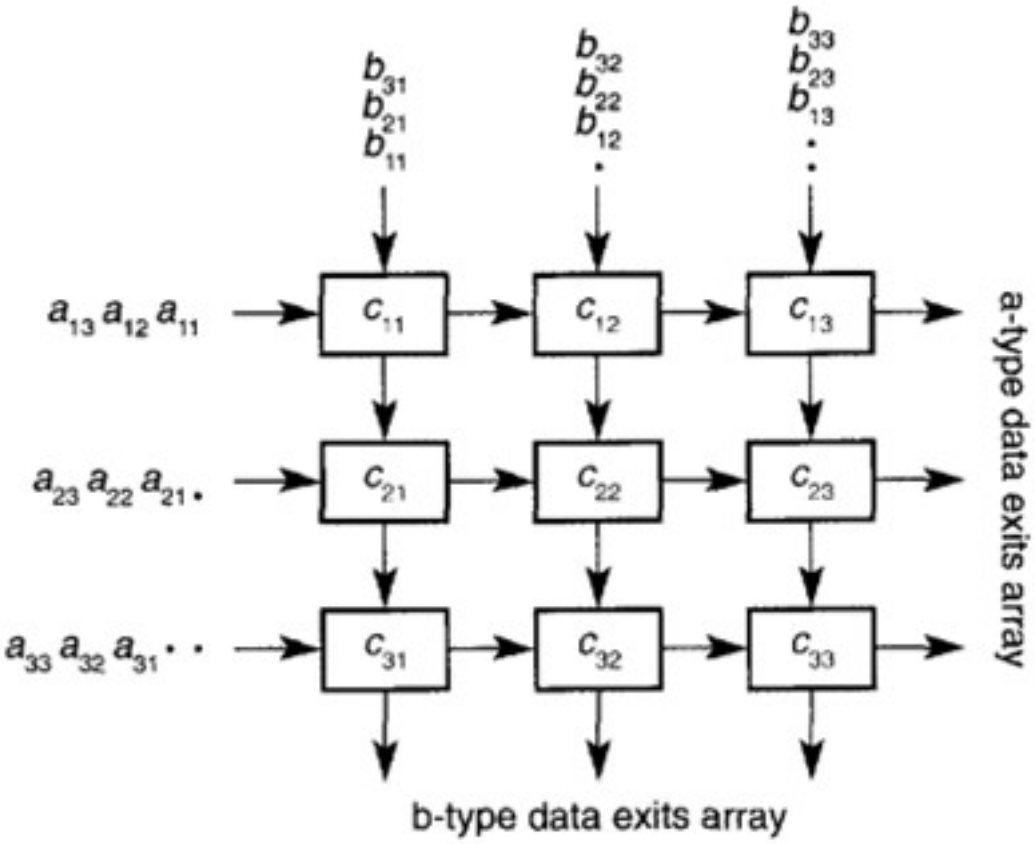
Máquinas sistólicas

- O mais restrito:
  - Ligados por uma rede fixa de interconexão;
  - Só os processadores da borda são acessíveis;
  - Memória local limitada;
  - Funcionamento síncrono
    - Recepção, Cálculo e Envio
  - Geralmente é implementado em CIs.

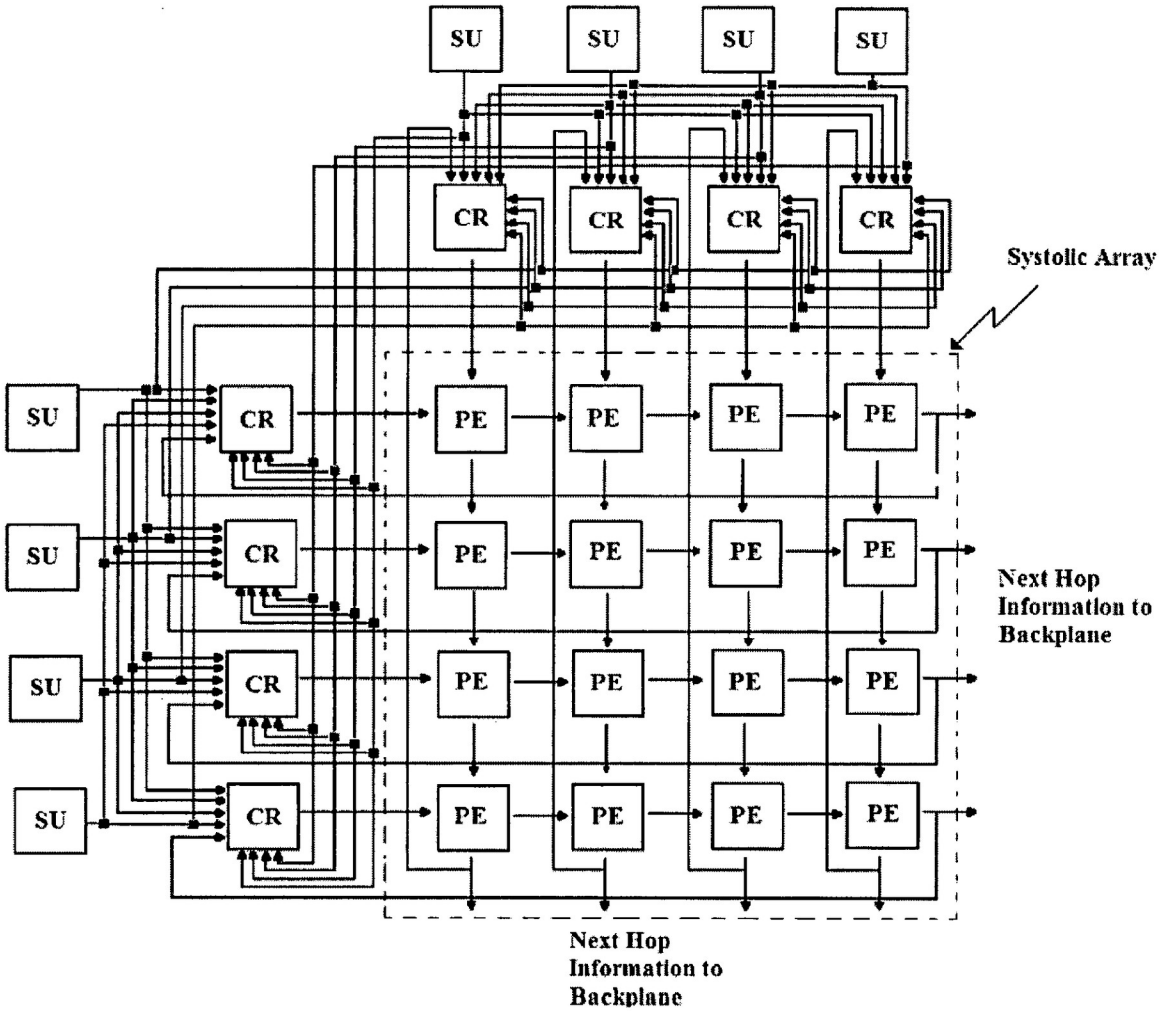
Exemplo sistólico



Exemplo sistólico



Ainda há interesse?



Modelos de execução

Modelos de execução

- Modelo usado no desenvolvimento
  - Predição de custos;
  - Planejamento da comunicação.

- Ligado ao modelo da máquina

Principais modelos

- PRAM
- Modelos com atraso de comunicação
- Modelo Sistólico
- Topologias específicas
- LogP
- BSP / CGM
- Tarefas maleáveis
- Futuro....

PRAM

- Parallel Random Access Machine
- Surgiu em 1978 e é uma referência
- Usado para análise de algoritmos paralelos

PRAM (características)

- Número ilimitado de processadores
- Síncrono
- Acesso a memória uniforme
- Memória ilimitada

PRAM (resolução de conflitos)

- EREW
- CREW
- CRCW
  - Regras para resolução de conflitos:
    - Arbitrária, Prioritária e Combinação

PRAM (variações)

- P-RAM
- BSR (difusão para toda a memória)
- XRAM (topologia dada)
- LPRAM (sem topologia)

- Mas a PRAM é muito poderosa, distante da realidade atual

PRAM (algoritmos)

- Soma de n elementos
- Soma Prefixa
- Ordenação

Descrição do Problema

thanks: Edson Cáceres

Dado um vetor A de n elementos A[0],...,A[n-1], a computação de prefixos calcula os valores:

A[0]

A[0] op A[1]

A[0] op A[1] op A[2]

...

A[0] op ... op A[n-1]

onde op é uma operação binária associativa.

- É também conhecido como Computacao de Prefixos

Descrição do Problema (cont.)

- Exemplo:

A operação binária é a adição.

vetor de entrada:

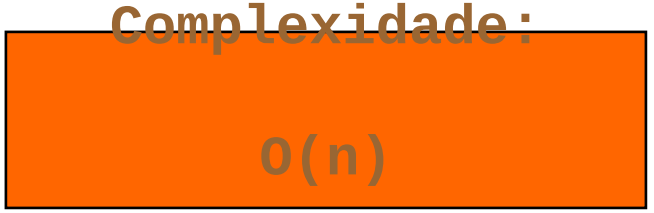
[3 7 5 1 2 4 0 9]

vetor resultante da soma de prefixos:

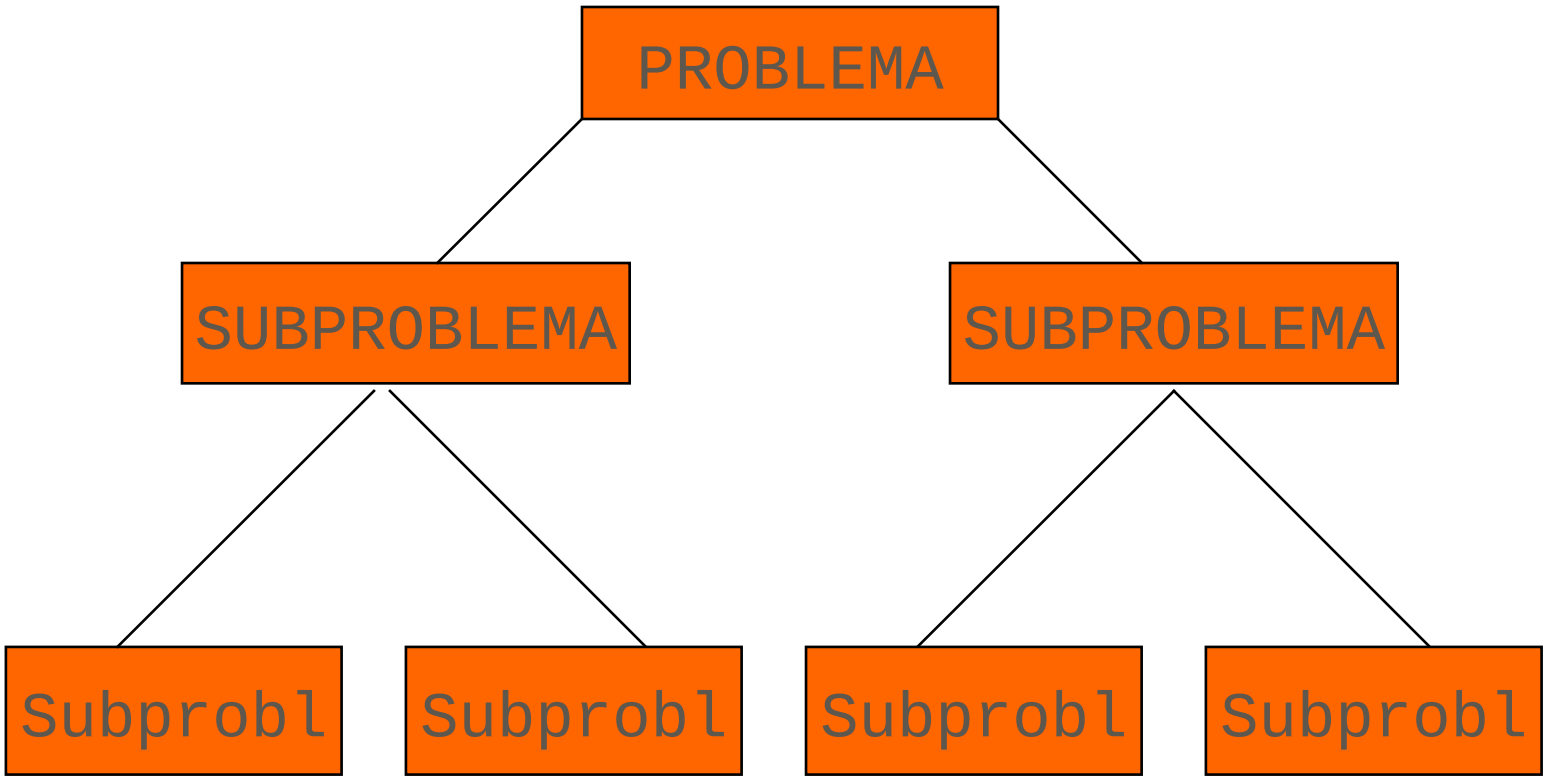
[3 10 15 16 18 22 22 31]

Algoritmo Sequencial

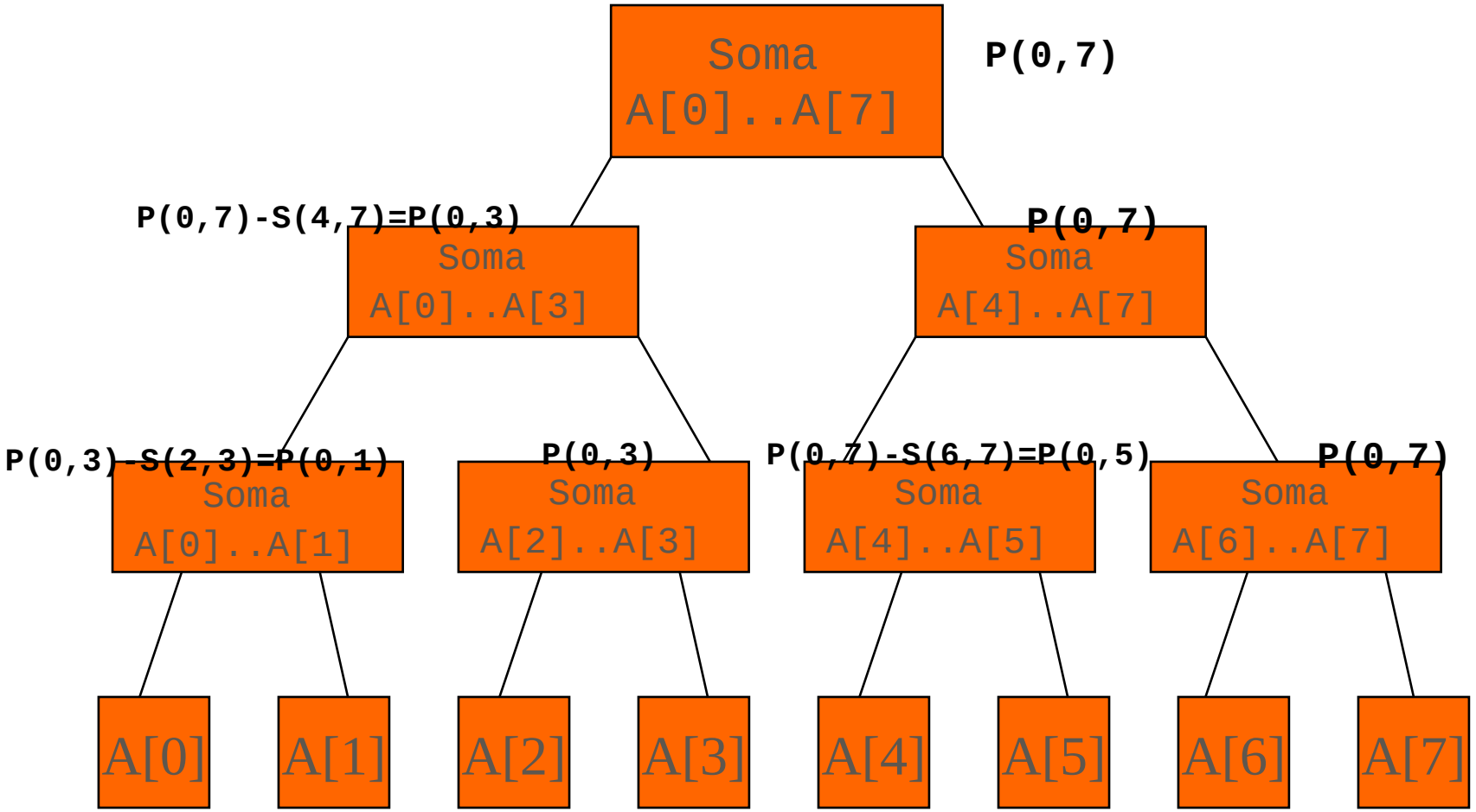
```
SOMA_PREFIXO(In, Out)
/* Passo 1: Out[0] recebe a soma do primeiro prefixo de In. Neste caso é
   o próprio In[0]. */
soma := In[0]
Out[0] := soma
/* Passo 2: Calcule os demais prefixos */
PARA i = 1 ATÉ i = n-1 FAÇA
    /* Passo 2.1. A soma do i-ésimo prefixo é a i-ésima posição do vetor
    de entrada mais soma.
    soma contém a soma do (i-ésimo)-1 prefixo. */
    Out[i] := In[i] + soma
    soma := Out[i]
```



Método da Árvore Binária Balanceada (cont.)



Apresentação do Algoritmo (cont.)



Modelos com atraso de comunicação

- Fornece um suporte para a comunicação
  - Processadores com memória local
- Vários resultados teóricos
- Considera o essencial das máquinas reais
  - Mas não considera

Grafo para os exemplos

Para ilustrar as diferenças dos diversos modelos usaremos:

Um único grafo

- Modelos
  - UET
  - UET - UCT
  - UET - LCT
  - SCT
  - Mais gerais

Algoritmos

- Grafo tipo send
- LCT com duplicação
- SCT com duplicação

Modelo Sistólico

- Exemplos:
  - Ordenação
  - Multiplicação de matrizes

Topologias específicas

- Muito usados no início da computação paralela;
- Motivação:
  - Ligações ponto a ponto
  - Encontrar formas de interconexão
- Objetivos:
  - Diminuir o diâmetro
  - Limitar o número de portas
  - Tolerância a falhas

Principais topologias

- Anel
- Toro
- Caminho
- Grade
- Hipercubo
- Mais sofisticadas
  - CCC (cube connected cycles)
  - De Bruijn

Formas de comunicação

- Tempo entre nós adjacentes
  - constante
  - modelo linear + $L\gamma$
  - modelo linear por pacotes
- Limitações
  - Full, ou Half duplex
  - número de portas 1,.., *

Vértices não adjacentes

- Existem as seguintes formas de transmissão
  - Store and Forward
  - Pipeline
  - Caminhos disjuntos
- Outras técnicas
  - wormhole
  - circuit-switching
  - + d + L

Algoritmos

- Geralmente restritos a padrões
  - Difusão
  - Difusão personalizada
  - Combinação
  - Troca completa
  - Troca completa personalizada
- Exemplos

Modelos realísticos

- Duas categorias:
  - Aproximam máquinas de forma genérica
    - LogP
  - Permitem algoritmos portáveis e eficazes
    - BSP / CGM
    - Tarefas maleáveis

LogP

- Número finito de processadores, custos de comunicação.
- L - Tempo de trânsito da mensagem
  - (pequena)
- o - custo adicional de envio e recepção
- g - intervalo entre envios/recepções seguidos
- P - número de processadores

LogP

- Os dados foram medidos em muitas máquinas reais;
- Pode-se prever o custo de programas com precisão;
- Existem extensões para mensagens grandes;

- É difícil obter resultados para grafos genéricos.

BSP e CGM

- Modelos semelhantes
  - Origens distintas:
  - BSP - Modelo programação
  - CGM - Modelo teórico

- Separação explícita do cálculo e da comunicação

BSP - princípios

- Super-Etapa
- Sincronização
- Parâmetros:
  - p - número de processadores
  - l - custo de sincronização
  - g - banda passante
  - comunicação: no máximo l/g-relação

CGM (coarse grained computers)

- Modelo teórico
- Dois parâmetros:
  - p - número de processadores
  - n - tamanho do problema
- Suposições:
  - n/p >> p
  - Memória local O(n/p)
  - n/p-relações
- Objetivo: minimizar a comunicação

Vantagens destes modelos

- CGM
  - Abstrai totalmente a máquina
  - Os algoritmos elaborados em CGM são bons
- BSP
  - Além de abstrair a máquina
  - Tem várias bibliotecas disponíveis
- Dica: Minicurso de 2002 (SBAC) CGM/BSP

Algoritmos

- Soma de n elementos
- Soma Prefixa

Tarefas maleáveis

- O paralelismo é implícito
- Cada tarefa pode ser executada em um ou mais processadores
  - Quanto maior o número maior a ineficiência
- Já existem resultados interessantes
  - Tarefas maleáveis independentes
  - Grafos de precedência

Computação em Grade

- Ligando vários computadores
- Ligando vários aglomerados

- Principal problema:
  - Heterogeneidade
    - Da rede
    - Dos computadores
- Modelos adequados:
  - BSP ou tarefas maleáveis

E tem mais

- Escalonamento dinâmico
- Tolerância a falhas
- Problemas com mais de um critério
  - Tempo de execução e energia