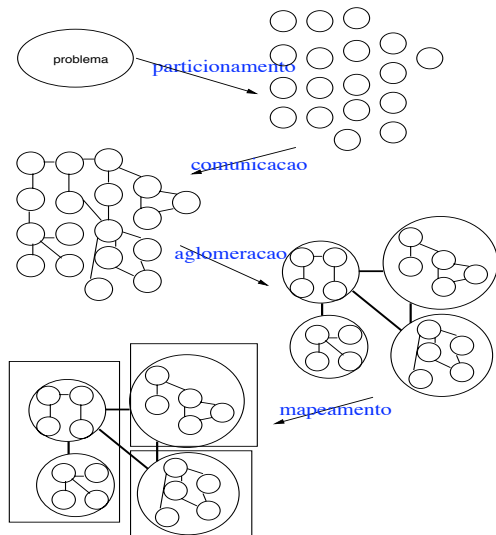


Projeto de Programas Paralelos (Ian Foster, SBPP)

Programação Concorrente e Paralela

Setembro de 2009





- idéia é expor oportunidades de paralelismo
 - foco em determinar uma grande quantidade de pequenas tarefas
- decomposição funcional e decomposição por domínio
 - exemplos

Verificação do Particionamento

- partição gera número de tarefas com ordem de magnitude superior à do número de processadores disponíveis?
 - flexibilidade para estágios posteriores
- tarefas são de tamanho comparável?
- número de tarefas cresce junto com tamanho da entrada do problema?
- há mais de uma alternativa de particionamento?



- como as tarefas definidas na fase 1 vão se comunicar?
 - definição de canais a serem usados
 - definição de mensagens
- (implementação pode não usar a abstração de canais...)
- alguns critérios de classificação são úteis:
 - local x global, estruturada x não estruturada, estática x dinâmica
 - assíncrona x síncrona

- comunicações *globais* podem criar gargalos de comunicação
 - exemplo da soma de N números
- comunicações *estruturadas* podem facilitar a tarefa de aglomerar e mapear
- sincronismo: simplicidade x menos oportunidades de concorrência

- todas as tarefas têm aproximadamente a mesma carga de comunicação?
 - distribuição ou replicação de estruturas de dados
- várias operações de comunicação podem ocorrer concorrentemente?
- e a computação? pode ocorrer concorrentemente?

- possibilidade de combinar tarefas
 - j pode levar em conta arquiteturas especficas
 - deixar o nmero de tarefas igual ao de processadores?
 - diminuico dos custos de comunicao e de troca de contexto
 - replicao de computaco

- objetivos podem ser conflitantes
 - granularidade \times custos de comunicao e criao de processos
 - n. tarefas $>$ processadores \Rightarrow possibilidade de sobrepor comunicao e computao
 - maior nmero de tarefas facilita balanceamento de carga

- diminumos a comunicaco?
- houve ganhos com replicaco de computaco?
- se replicamos dados, isso no compromete a escalabilidade?
 - ... o nmero de tarefas ainda cresce com o tamanho do problema?
- ainda h concorrncia suficiente?

Mapeamento

- 1 colocar processos que podem executar concorrentemente em processadores diferentes
 - 2 colocar processos que se comunicam com frequência no mesmo processador
 - ... claramente pode haver conflitos
- problema de mapeamento é NP-completo
 - na prática muitas vezes situações mais simples...



Mapeamento (e Balanceamento de Carga)

- escalonador procura alocar processos da melhor maneira possível
- alterações durante a execução podem tornar inviável um bom mapeamento inicial
 - dificuldade de prever uso de recursos de cada processo
 - criação dinâmica de novos processos
 - variação na carga externa à aplicação
- dificuldades com balanceamento
 - uso de recursos para determinar estado de execução de processadores
 - complexidade de transferir de processos em execução



- analisamos diferentes estratégias de criação de processos?
 - SPMD x criação dinâmica de processos
- se usamos um esquema de balanceamento de carga com um gerente central, será que ele não irá se tornar um gargalo?
- verificamos os custos do algoritmo de balanceamento de carga?

Exemplo: temperatura numa vara

- cálculo da evolução da temperatura ao longo do tempo
 - matriz de pontos: posição x tempo



Como paralelizar

- 1 particionamento
 - cada ponto da matriz é uma tarefa
- 2 comunicação
 - cálculo em cada ponto depende do resultado de outros três
- 3 aglomeração e mapeamento
 - não há como trabalhar em paralelo nas “colunas”
 - aglomeração de algumas posições

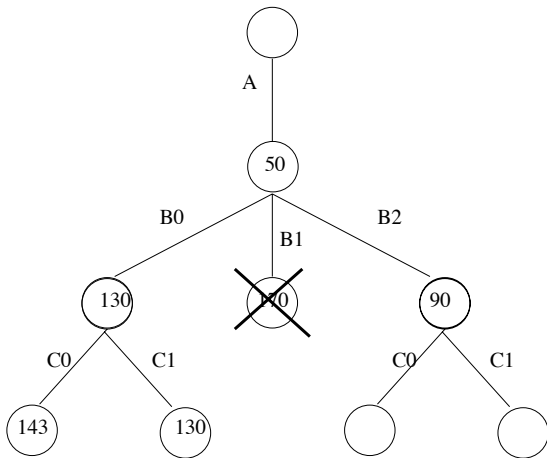


Exemplo: projeto de componentes

- n células têm que ser dispostas em uma placa
- existem restrições em relação às posições relativas
- exploração de todas as soluções tem custo proibitivamente alto
- uso de estratégia *branch and bound*



Branch and Bound



- particionamento: criação de processos para buscas paralelas
 - em primeira abordagem uma por nó...
- comunicação: balanço entre atualização do mínimo encontrado e economia de comunicação
 - gerente central pode manter soluções e mínimo
 - escalável?

- aglomeração: um único processo pode explorar sub-árvore
- mapeamento: alternativas
 - único processo raiz distribui tarefas para trabalhadores
 - diversos processos replicam trabalho inicial e assumem cada um uma sub-árvore a partir de algum ponto