Influência das Estratégias de Particionamento de Dados no Desempenho de Aplicações Numéricas Paralelas

Claudio Schepke, Tiarajú Asmuz Diverio, Nicolas Maillard

Mestrado em Ciência da Computação
Programa de Pós-Graduação em Computação
Universidade Federal do Rio Grande do Sul - Brasil

{cschepke,diverio,nicolas}@inf.ufrgs.br

Apoio: CNPq

WSPPD'2007

Sumário

Introdução

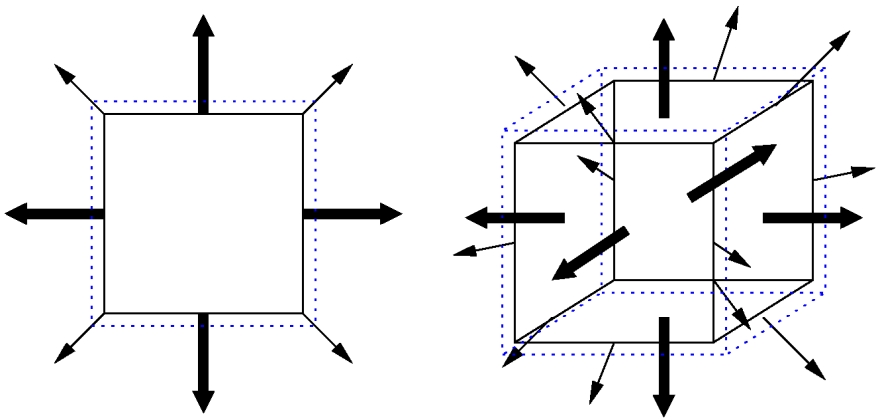
- Uso de implementações paralelas em aplicações numéricas
  - Tempo computacional inaceitável
  - Recursos de memória
  - Limites de tempo
- Paralelismo de Dados
- Distribuição dos dados
  - Fatiamento por número de processadores
  - Fatiamento cíclico
- Particionamento em Blocos
  - Usado na resolução de sistemas lineares
  - Melhor distribuição e acesso aos dados em memória
  - Taxa de acerto de *cache*
  - Equilíbrio de carga e dependência de dados

Objetivos do Trabalho

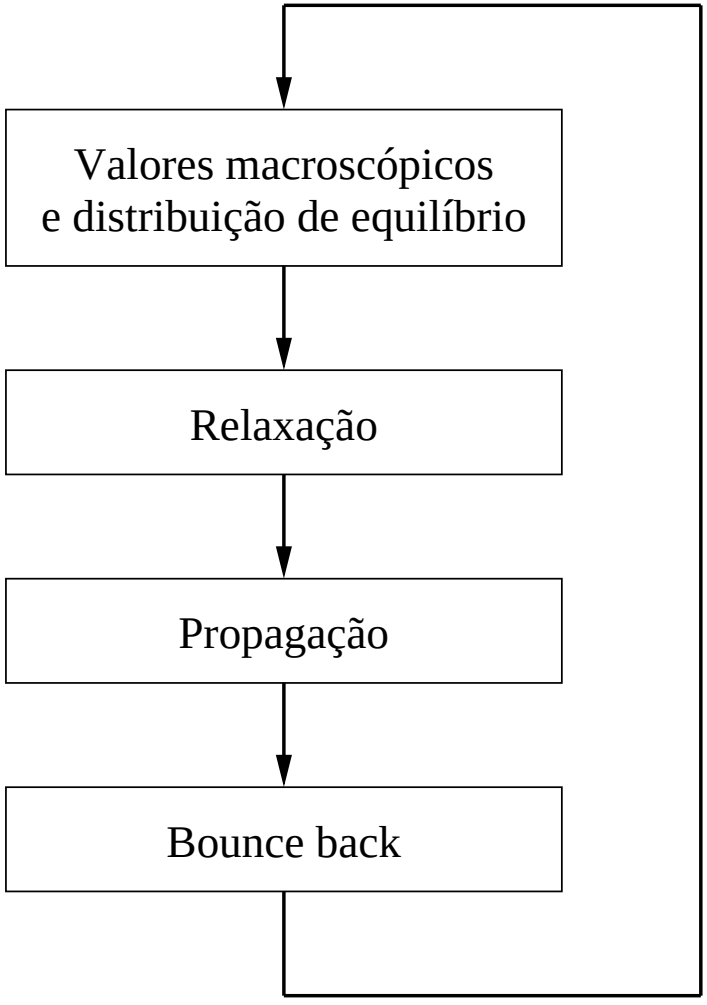
- Analisar o desempenho paralelo de diferentes formas de particionamento de dados
  - Avaliar o uso de distribuição de dados em blocos
  - Implementações paralelas do Método de Lattice Boltzmann
  - Diferentes configurações de mapeamento dos dados

Método de Lattice Boltzmann

- Método numérico iterativo e discreto para a modelagem e simulação mesoscópica de fluxos de fluidos
- Definido essencialmente por funções de propagação e relaxação como forma de representar o fluxo das partículas e o cálculo dos valores físicos
- Modelos de reticulado
  - D2Q9 - Bidimensional e 9 direções de deslocamento
  - D3Q19 - Tridimensional e 19 direções de deslocamento



Algoritmo

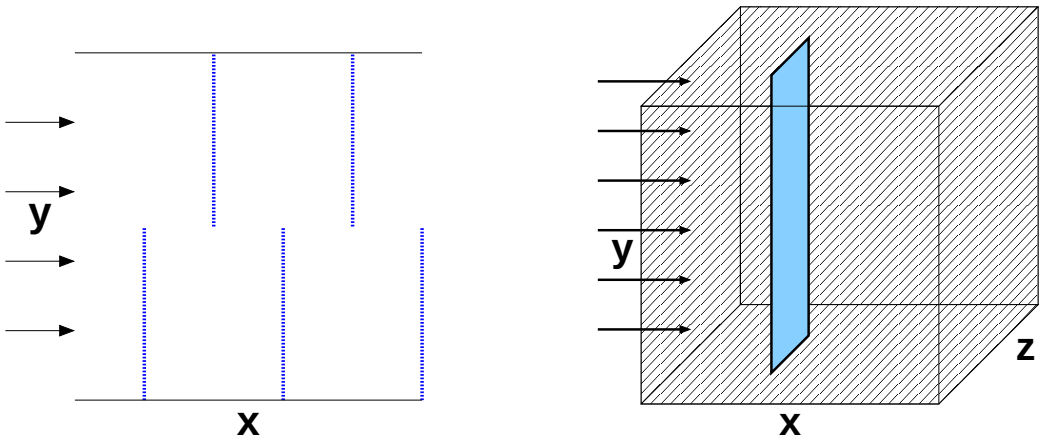

Valores macroscópicos
e distribuição de equilíbrio
Relaxação
Propagação
Bounce back

Implementação e Paralelização

- Implementação
  - Linguagem de programação C
  - Modelo de programação SPMD
  - Comunicação interprocessos via MPI
- Paralelização
  - Execução concorrente dos processos sobre os dados do reticulado
  - Divisão dos dados do fluido em blocos, segundo o número de processadores disponível, usando mapeamento cartesiano de MPI
  - Troca de dados entre os subreticulados vizinhos ao final de cada iteração

Avaliações de Desempenho

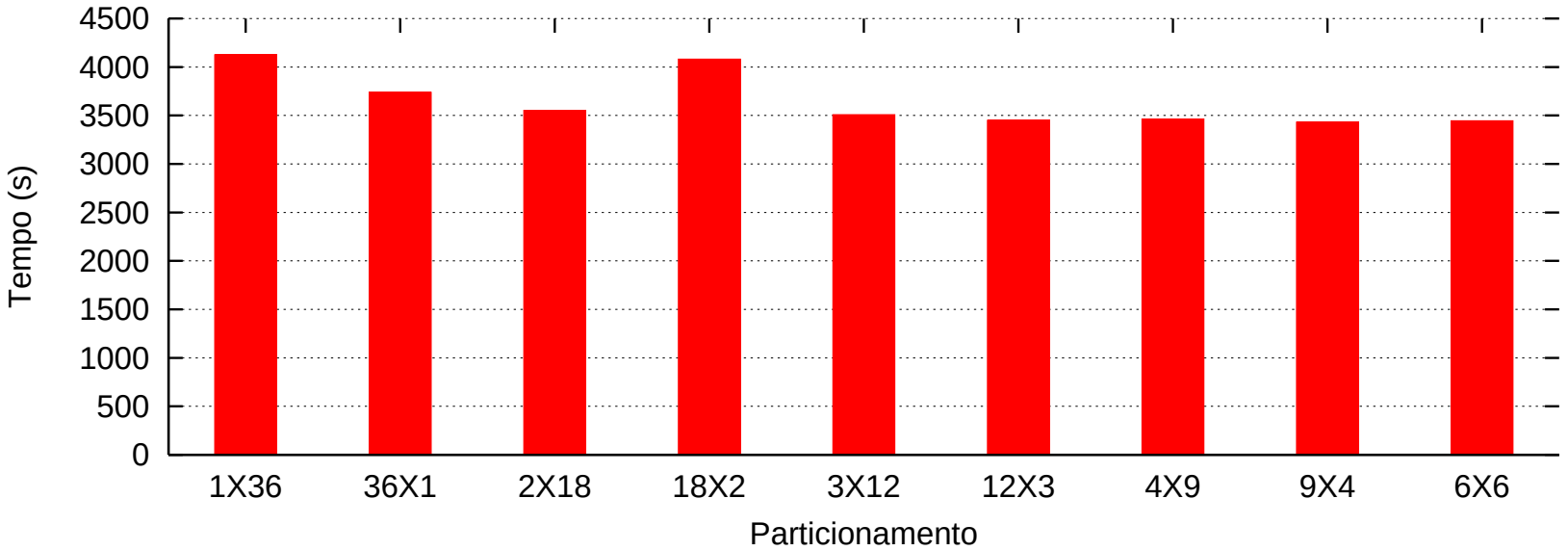
- Estudo de casos
  - Reticulado bidimensional de 512 $\times$ 512 elementos
  - Reticulado tridimensionai de 128 $\times$ 128 $\times$ 128 elementos
  - Canais com obstáculos



- Obtenção dos resultados experimentais
  - Execuções no cluster labtec
  - Registro de tempo usando a função *MPI_Wtime*
  - 10 execuções - exclusão do melhor e pior resultado
  - Desvio padrão inferior a 1% de tempo entre as execuções

Resultados - Bidimensional

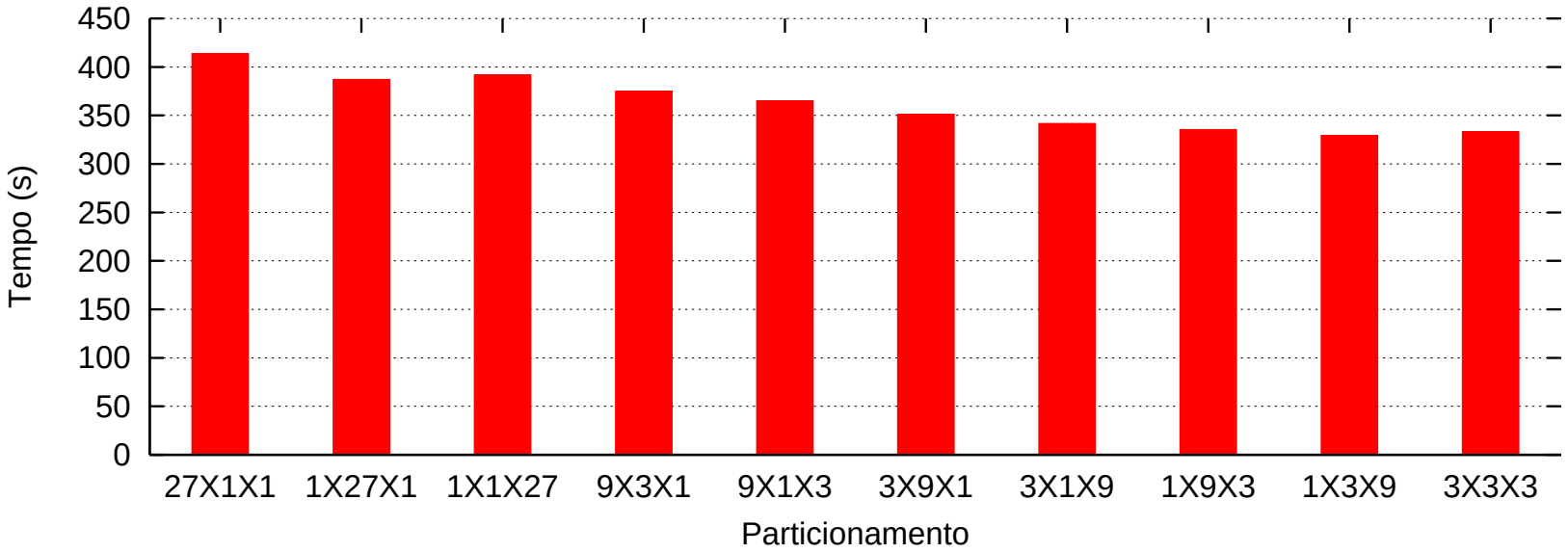
- Tempo de execução após 150.000 iterações utilizando 36 processadores



- Ganho de desempenho de até 8,23% em relação a melhor execução linear

Resultados - Tridimensional

- Tempo de execução após 447 iterações utilizando 27 processadores



- Ganho de desempenho de até 14,95% em relação a melhor execução linear

Considerações Finais

- Análise dos resultados
  - Subreticulados mais quadrados ou cúbicos apresentaram resultados melhores
  - Ganho de desempenho significativo em relação ao particionamento unidimensional
  - Resultados e comportamento semelhantes para outras configurações e número de processadores usados
- Conclusão
  - Importância do uso de estratégias de particionamento de dados eficientes
  - Particionamento em Blocos como forma de aumentar o desempenho das aplicações numéricas paralelas
  - Influência do custo de comunicação em aplicações com dependência de dados