Universidade Estadual de Campinas
Instituto de Computação

MO644 - Introdução à Programação Paralela

Projeto Final

Alunos:

147512, Nathália Menini Cardoso dos Santos

192744, Miguel Antonio Rodriguez Santander

Professor:

Guido Araújo

XX de XXXX de 2017

Resumo

Processos que utilizam matrizes são muito comuns em diversas áreas do conhecimento. O tempo de computação destes métodos está diretamente relacionado com a dimensão de tais estruturas matemáticas, assim, resolver os problemas de grandes matrizes significa um maior tempo de processamento. Um problema simples, porém muito recorrente, pode ser a resolução de um sistema linear, em que o número de operações pode se tornar relativamente grande, dependendo da aplicação em questão. No âmbito desse problema, alguns métodos como a Fatoração de Cholesky são adotados. Nessa linha, temos a intenção de fazer uma comparação exaustiva do método de Cholesky usando algumas ferramentas estudadas em aula (OpenMP, Pthreads e CUDA). Esta comparação permitirá observar e analisar como o desempenho do algoritmo pode ser melhorado quando implementado em paralelo.

1 Introdução

2 Fatoração de Cholesky

Sejam $A \in \mathbb{M}_n(\mathbb{R})$ uma matriz positiva-definida e um elemento $b \in \mathbb{R}^n$. Consideremos o problema de encontrar $x^* \in \mathbb{R}^m$ solução do sistema linear positivo-definido

$$Ax = b.$$

Podemos obter uma solução numérica através da Fatoração de Cholesky da matriz A, garantida pelo teorema abaixo, cuja demonstração pode ser encontrada em [1].

Teorema *Seja $A \in \mathbb{M}_n(\mathbb{R})$ uma matriz positiva definida. Então, existe uma única matriz triangular superior G, com os elementos da diagonal principal positivos, tal que $A = G^tG$.*

Em Algorithm 1 mostramos o algoritmo serial de Cholesky (retirado de [2]). Após uma rigorosa análise, percebemos que o algoritmo está calculando iterativamente as linhas de G^t (indicada no algoritmo por L), de modo que a linha $i+1$ depende dos resultados obtidos na iteração i, gerando assim um *loop* DOACROSS. Além disso, no laço interno temos que o valor de $j+1$ depende do resultado de j, gerando um *loop carried dependence*. Entretando, se trocarmos a ordem dos laços referentes as linhas e colunas (i e j) e anticiparmos o calculo da diagonal principal, não teríamos mais dependencia entre o *loop* interno (agora em i), porém, o laço externo continua não paralelizável. Em Algorithm 2 exibimos esse novo código serial, que agora pode ser paralelizado. Portanto, daqui para frente, utilizaremos sempre o Algorithm 2.

for $i = 0, ..., n-1$ **do**
  for $j = 0, ..., i$ **do**
    double $s = 0.0$;
    for $k = 0, ..., j-1$ **do**
      $s = s + L[i * n + k] * L[j * n + k]$;
    end
    if $i == j$ **then**
      $L[i * n + j] = \text{sqrt}(A[i * n + i] - s)$;
    else
      $L[i * n + j] = 1/L[j * n + j] * (A[i * n + j] - s)$;
    end
  end
end

Algorithm 1: Algoritmo serial não paralelizável da Fatoração de Cholesky

for $j = 0, ..., n-1$ **do**
  double $s = 0.0$;
  for $k = 0, ..., j-1$ **do**
    $s = s + L[j * n + k] * L[j * n + k]$;
  end
  $L[j * n + j] = \text{sqrt}(A[j * n + j] - s)$;
  for $i = j+1, ..., n-1$ **do**
    double $s = 0.0$;
    for $k = 0, ..., j-1$ **do**
      $s = s + L[i * n + k] * L[j * n + k]$;
    end
    $L[i * n + j] = 1/L[j * n + j] * (A[i * n + j] - s)$;
  end
end

Algorithm 2: Algoritmo serial paralelizável da Fatoração de Cholesky

3 *Profile*

4 Descrição da paralelização

5 Resultados

6 Referências Bibliográficas

[1] Pulino, P. *"Álgebra Linear e suas Aplicações: Notas de Aula"*. (2004)

[2] `https://rosettacode.org/wiki/Cholesky_decomposition#C` [Acessado em 22/05/2017]