

## Prova 1 - PCAM

João Pedro A.S. Secundino (10692054); João Pedro Uchôa Cavalcante (10801169); Luís Eduardo Rozante de Freitas Pereira (10734794); Sérgio Ricardo G. B. Filho (10408386);

### 1. Design

#### 1.0.1 P - PARTICIONAMENTO

Sejam A, B, C, D matrizes  $N \times N$  (sem perda de generalidade). Para a multiplicação das matrizes, cada tarefa será responsável por multiplicar dois valores, sendo cada um deles proveniente de uma das matrizes operando. Por exemplo, para realizar a multiplicação  $A * B$ , seriam criadas  $N^3$  tarefas  $T_{ijk}$ , ( $i, j, k \in [1, N]$ ), cada uma responsável por multiplicar os valores  $A[i][j] * B[j][k]$ . Portanto, como serão realizadas duas multiplicações de matrizes em paralelo, serão criadas  $2 * N^3$  tarefas  $T_{1ijk}$  e  $T_{2ijk}$ . As tarefas  $T_{xijk}$ , ( $x \in [1, 2]$ ) relacionadas a um mesmo elemento resultante deverão se comunicar por meio de uma redução de forma a calcular o valor acumulado deste elemento ( $R_{AB_{ij}}$  ou  $R_{CD_{ij}}$ ). Assim que as tarefas realizarem esta computação, elas devem comunicar os seus resultados às tarefas que serão criadas a seguir para que o programa possa realizar a soma de matrizes.

A partir de agora, serão criadas  $N^2$  tarefas para a realização da soma de matrizes. Cada tarefa será responsável por somar dois valores  $R_{AB_{ij}}$  ou  $R_{CD_{ij}}$ , sendo cada um proveniente das multiplicações de matrizes realizadas anteriormente. Após realizada a computação, as tarefas  $T_{ij}$  gerarão  $N^2$  resultados, que deverão ser comunicados à tarefa mestre para que a saída do programa seja gerada.

#### 1.0.2 C - COMUNICAÇÃO

Para que a multiplicação de matrizes seja possível, cada tarefa  $T_{xijk}$  deverá receber como entrada os dois elementos a serem multiplicados provenientes das matrizes de entrada. Outra comunicação a ser realizada acontece quando as tarefas relacionadas a um mesmo elemento resultante  $R_{AB_{ij}}$  ou  $R_{CD_{ij}}$  devem acumular os seus resultados. Os canais pelos quais essa comunicação será realizada poderão ser duas estruturas auxiliares globais  $R_{AB}[i][j]$  e  $R_{CD}[i][j]$  necessárias para a acumulação desses resultados. Assim que os seus valores forem armazenados, a tarefa deixam de existir.

Na etapa da soma, que deverá ser realizada após uma sincronização, teremos duas matrizes resultantes  $R_{AB}$   $R_{CD}$  de dimensão  $N \times N$ . Essas tarefas  $T_{ij}$  necessitam, para iniciar as suas computações, dos dois valores a serem somados  $R_{AB}[i][j]$  e  $R_{CD}[i][j]$ . Para comunicar os seus resultados à tarefa mestre, o canal a ser utilizado será uma matriz global  $R_f$  de dimensão  $N \times N$ .

### 1.0.3 A - AGLOMERAÇÃO

Para reduzir o *overhead* de comunicação e maximizar a localidade, as tarefas relacionadas à blocos de linhas das matrizes de entrada serão aglomeradas, de forma a distribuir estas tarefas em P processos. Por exemplo, as tarefas relacionadas às P primeiras linhas das matrizes de entrada serão executadas pelo primeiro processo, enquanto as relacionadas às P próximas linhas serão executadas pelo segundo processo e assim por diante. Matematicamente, as tarefas  $T_{xijk}(i \in [1, P])$  serão executadas pelo processo  $P_1$ ,  $T_{xijk}(i \in [P + 1, 2P])$  pelo  $P_2$  e assim por diante. Assim, serão formados P processos acumulando  $\frac{2*N^3}{P}$  tarefas com computações equivalentes. No caso da divisão de tarefas não ser exata, as tarefas restantes serão distribuídas seguindo uma fila circular.

Após realizada a sincronização, a aglomeração da segunda etapa seguirá a mesma ideia da anterior, ao dividir as linhas das matrizes de entrada em P processos, realizando, em cada um deles,  $\frac{N^2}{P}$  computações equivalentes. Assim, as tarefas  $T_{ij}(i \in [1, P])$  serão realizadas por  $P_1$ , as  $T_{ij}(i \in [P + 1, 2P])$  por  $P_2$  e assim por diante. No caso da divisão não ser exata, as tarefas restantes serão distribuídas seguindo uma fila circular.

### 1.0.4 M - MAPEAMENTO

De acordo com Ian Foster<sup>1</sup>, o problema do mapeamento não acontece em sistemas que possuem memória compartilhada e oferecem escalonamento automático de tarefas. Portanto, nesta atividade, como o computador no qual o programa será executado possui as características citadas, o problema do mapeamento não será abordado durante a implementação do algoritmo.

---

1. FOSTER, Ian. Designing and building parallel programs: concepts and tools for parallel software engineering. Addison-Wesley Longman Publishing Co., Inc., 1995.