Advanced Numerical Methods September 21, 23 and 30, 2011

Lecture Notes 2: Parallel matrix multiplication

Lecturer: Che-Rung Lee *Scribe: Po-Yi Hsu 100062647*

1.3 Parallel matrix multiplication

1.3.1 Mesh parallel architecture

- As shown in Figure 1, processors can only communicate with their neighbors in mesh architecture.

- SUMMA algorithm

  Let $A = \begin{pmatrix} \vec{a_1} & \vec{a_2} & \vec{a_3} & \vec{a_4} \end{pmatrix}$, $B = \begin{pmatrix} \vec{b_1}^T \\ \vec{b_2}^T \\ \vec{b_3}^T \\ \vec{b_4}^T \end{pmatrix}$.

- Then $C = A \cdot B = \sum_{i=1}^{4} \vec{a_i}\vec{b_i}^T$.

- Each iteration A broadcasts $\vec{a_i}$ along row and B broadcasts $\vec{b_i}$ along column, and processors will multiply them separately.

- At i iteration, we could get $\vec{a_i}\vec{b_i}^T$. So after 4 iterations we can get $\vec{a_1}\vec{b_1}^T + \vec{a_2}\vec{b_2}^T + \vec{a_3}\vec{b_3}^T + \vec{a_4}\vec{b_4}^T$ for the answer.

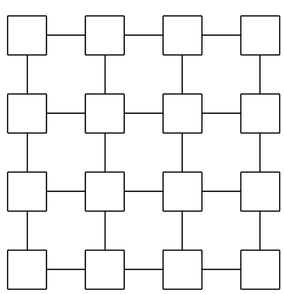


Figure 1: Mesh parallel architecture

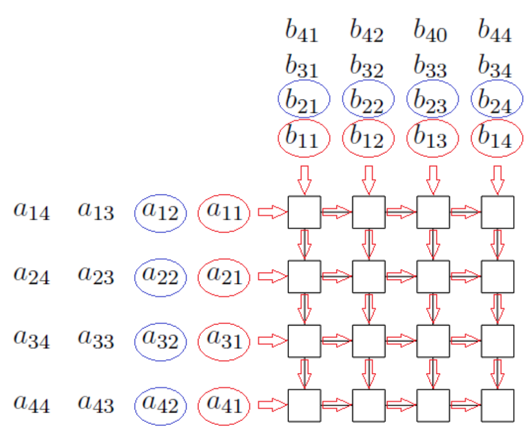
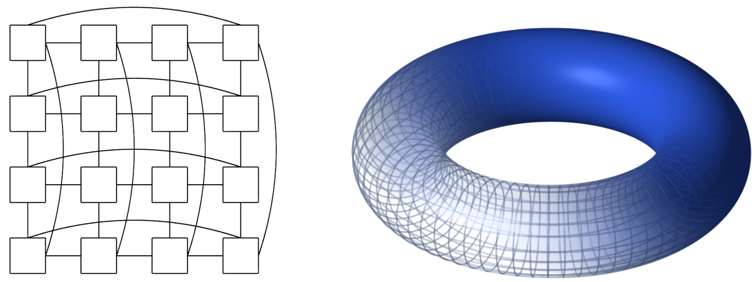


Figure 2: SUMMA algorithm

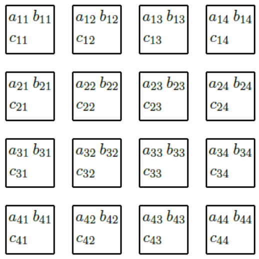


(a) Mesh with feedback lines (b) Torus example

Figure 3: Torus parallel architecture

1.3.2 Torus parallel architecture

- Torus architecture adds feedback lines to mesh architecture, as shown in Figure 3.
- Cannon's algorithm
  Let $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$, $B = \begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{pmatrix}$
  Then $C = A \cdot B \begin{pmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \\ c_{41} & c_{42} & c_{43} & c_{44} \end{pmatrix}$
- We want each processor node to compute a c_{ij} in parallel.
- How to decide which node to compute which matrix elements?
- First, we place matrix elements as in Figure 4(a).
- As the algorithm progresses, the elements of A are passed left and elements of B are passed upward.
- We want elements are moved from their initial positions to align them so that the correct elements are multiplied with one another.
- From Figure 4(b), we observe some regularity. So we rearrange matrix elements as in Figure 5 and place them into processor nodes in Figure 6.
- Now at the first iteration, we could get multiplied elements of black cycles in Figure 4(b) whose elements are at the initial place in Figure 6.
- At the second iteration, as elements shifted by the colored arrow in Figure 6, we could get multiplied elements of same color cycles in Figure 4(b).



(a) Matrix elements placement

$$
\begin{aligned}
c_{11} &= a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} + a_{14}b_{41} \\
c_{12} &= a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} + a_{14}b_{42} \\
c_{13} &= a_{11}b_{13} + a_{12}b_{23} + a_{13}b_{33} + a_{14}b_{43} \\
c_{14} &= a_{11}b_{14} + a_{12}b_{24} + a_{13}b_{34} + a_{14}b_{44} \\
c_{21} &= a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} + a_{24}b_{41} \\
c_{22} &= a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} + a_{24}b_{42} \\
&\vdots
\end{aligned}
$$

(b) Equations of c_{ij}

Figure 4: Try to find out some regularity

- Cannon's algorithm minimizes the communication among processor nodes.
- We also can replace matrix elements with matrix blocks.

$$
A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \quad B = \begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{pmatrix}
$$

Figure 5: Rearrange matrix A and B

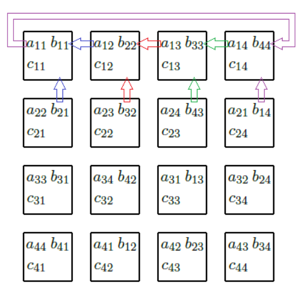


Figure 6: New matrix elements placement