Parallel Random Number

Tuesday, October 16, 2018 11:20 AM

Detailed Parallel Algorithm:

step 1) - Load input parameters $\{x_0, A, B, P, n\}$ on all procs.

Step 2) At each process P_j:

a) Initialize: Matrix $M = \begin{bmatrix} A & 0 \\ B & 1 \end{bmatrix}$, $M^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

b) Initialize: - Array X_{local} of size n/p, with each element of type a 2x2 Matrix.

- Init: $X_{local} = [\underbrace{M \; M \; \ldots \; M}_{n/p}]$ (indices 0 to $n/p - 1$)

Step 3) At each process P_j:

- $M_{local} = M^0$
- for $(i = 0$ to $n/p - 1)$
  - $M_{local} = M_{local} \otimes X_{local}[i]$

Parallel Random Number

Tuesday, October 16, 2018 11:20 AM

Step 4) • Run p-element parallel prefix, with each process providing its correspond M_{local} as input, and $\otimes$ as the matrix Multiplication operator.

• Output will be a 2x2 matrix, say M_{off} which represents the prefix matrix product.

M_{off} at process P_j will be equal to: $M^{j\frac{n}{p}}$

Step 5) At every process P_j:

- Call serial_matrix $(\frac{n}{p})$, with two modifications:
  - Initialize its $M\_next \leftarrow M_{off}$
  - Run its for loop from 0 to n-1

Output X from every process.