Prefix Sums Algorithm on a Hypercube Architecture

Introduction

The prefix sum algorithm enables each processor to compute the cumulative sum of elements up to its rank:

$$\text{prefix}_i = x_0 + x_1 + \cdots + x_i$$

where $x = [x_0, x_1, \ldots, x_{n-1}]$.

For this example, we consider $n = 8$, working in a 3-dimensional hypercube architecture. Each processor has a unique 3-bit id (rank) and is connected to processors differing by exactly one bit. This connection pattern provides each processor with $d = \log_2 8 = 3$ neighbors. Each processor p, with $p = 0, \ldots, 7$, initially holds a value x_p from the input array x.

Algorithm Steps

The algorithm proceeds in $\log_2 n$ steps, where at each step, processors exchange messages representing partial sums with specific neighboring processors. Each processor also updates its local sum, but only *if it receives a message from a processor with a lower rank.*

1. **Initialization**: Each processor p stores its respective element x_p and initializes its current sum as $\text{sum}_p = x_p$.

   Initial locally stored messages:

$$\begin{array}{ll} \text{Processor } 0 : x_0 & \text{Processor } 1 : x_1 \\ \text{Processor } 2 : x_2 & \text{Processor } 3 : x_3 \\ \text{Processor } 4 : x_4 & \text{Processor } 5 : x_5 \\ \text{Processor } 6 : x_6 & \text{Processor } 7 : x_7 \end{array}$$

   Initial cumulative sums:

$$\begin{array}{ll} \text{sum}_0 = x_0 & \text{sum}_1 = x_1 \\ \text{sum}_2 = x_2 & \text{sum}_3 = x_3 \\ \text{sum}_4 = x_4 & \text{sum}_5 = x_5 \\ \text{sum}_6 = x_6 & \text{sum}_7 = x_7 \end{array}$$

2. **Step 1 (Neighbor distance of 1)**: Each processor p sends its current message to the processor differing by 1 in the least significant bit (exchanges between processors 0-1, 2-3, 4-5, 6-7). Upon receiving the message, each processor updates its stored message to prepare for the next step. Processors with a lower-ranked neighbor also add the received message to their local sum.

   Locally stored messages after Step 1:

$$\begin{array}{ll} \text{Processor } 0 : x_0 + x_1 & \text{Processor } 1 : x_0 + x_1 \\ \text{Processor } 2 : x_2 + x_3 & \text{Processor } 3 : x_2 + x_3 \\ \text{Processor } 4 : x_4 + x_5 & \text{Processor } 5 : x_4 + x_5 \\ \text{Processor } 6 : x_6 + x_7 & \text{Processor } 7 : x_6 + x_7 \end{array}$$

   Cumulative sums after Step 1:

$$\begin{aligned} \text{sum}_0 &= x_0 \\ \text{sum}_1 &= x_0 + x_1 \\ \text{sum}_2 &= x_2 \\ \text{sum}_3 &= x_2 + x_3 \\ \text{sum}_4 &= x_4 \\ \text{sum}_5 &= x_4 + x_5 \\ \text{sum}_6 &= x_6 \\ \text{sum}_7 &= x_6 + x_7 \end{aligned}$$

3. **Step 2 (Neighbor distance of 2)**: Each processor p sends its message to the processor differing by 1 in the second least significant bit (exchanges between processors 0-2, 1-3, 4-6, 5-7). After receiving the message, each processor updates its stored message to prepare for the next step. Processors with lower-ranked neighbors also add the received message to their local sum.

   Locally stored messages after Step 2:

$$\begin{array}{ll} \text{Processor } 0 : x_0 + x_1 + x_2 + x_3 & \text{Processor } 1 : x_0 + x_1 + x_2 + x_3 \\ \text{Processor } 2 : x_0 + x_1 + x_2 + x_3 & \text{Processor } 3 : x_0 + x_1 + x_2 + x_3 \\ \text{Processor } 4 : x_4 + x_5 + x_6 + x_7 & \text{Processor } 5 : x_4 + x_5 + x_6 + x_7 \\ \text{Processor } 6 : x_4 + x_5 + x_6 + x_7 & \text{Processor } 7 : x_4 + x_5 + x_6 + x_7 \end{array}$$

Cumulative sums after Step 2:

$$
\begin{aligned}
\text{sum}_0 &= x_0 \\
\text{sum}_1 &= x_0 + x_1 \\
\text{sum}_2 &= x_0 + x_1 + x_2 \\
\text{sum}_3 &= x_0 + x_1 + x_2 + x_3 \\
\text{sum}_4 &= x_4 \\
\text{sum}_5 &= x_4 + x_5 \\
\text{sum}_6 &= x_4 + x_5 + x_6 \\
\text{sum}_7 &= x_4 + x_5 + x_6 + x_7
\end{aligned}
$$

4. **Step 3 (Neighbor distance of 4)**: Each processor p sends its message to the processor differing by 1 in the most significant bit (exchanges between processors 0-4, 1-5, 2-6, 3-7). Upon receiving the message, each processor updates its stored message to prepare for the next step. Processors with a lower-ranked neighbor also add the received message to their local sum.

   Locally stored messages after Step 3:

$$
\begin{array}{ll}
\text{Processor } 0 : \sum_{i=0}^{7} x_i & \text{Processor } 1 : \sum_{i=0}^{7} x_i \\
\text{Processor } 2 : \sum_{i=0}^{7} x_i & \text{Processor } 3 : \sum_{i=0}^{7} x_i \\
\text{Processor } 4 : \sum_{i=0}^{7} x_i & \text{Processor } 5 : \sum_{i=0}^{7} x_i \\
\text{Processor } 6 : \sum_{i=0}^{7} x_i & \text{Processor } 7 : \sum_{i=0}^{7} x_i
\end{array}
$$

   Cumulative sums after Step 3 — Prefix sums:

$$
\begin{aligned}
\text{sum}_0 &= x_0 \\
\text{sum}_1 &= x_0 + x_1 \\
\text{sum}_2 &= x_0 + x_1 + x_2 \\
\text{sum}_3 &= x_0 + x_1 + x_2 + x_3 \\
\text{sum}_4 &= x_0 + x_1 + x_2 + x_3 + x_4 \\
\text{sum}_5 &= x_0 + x_1 + x_2 + x_3 + x_4 + x_5 \\
\text{sum}_6 &= x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \\
\text{sum}_7 &= x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7
\end{aligned}
$$