Hossein Souri

EN 601.320/420/620
November 19, 2020

Project 4

Questions

Q1- Compare the horizontal data decomposition that you implemented with a 2-d geometric decomposition (as in the Dask workbook). This decomposition divides the grid recursively into smaller squares, e.g. an n x n grid may consist of 4 (n/2) x (n/2) squares or 16 (n/4) x (n/4) squares or 64 (n/8) x (n/8) squares. Give exact counts as a function of n (the size of the input) and p (the number of processes).

Answer:
In our implementation we used horizontal decomposition in which we just need to send the first and last row to other processors. In the 2-d decomposition we should communicate with all eight neighbors. However, the length of messages are smaller compared to the horizontal case. In general, an $n \times n$ grid is consisted of p squares of size $(\frac{n}{\sqrt{p}} \times \frac{n}{\sqrt{p}})$.

What is the total number of messages sent in each decomposition?
Answer:

For the horizontal decomposition, if we send a row or a column entirely through one message, then we need to send 2 messages to the top and bottom nodes. So in total we should send out $2 \times p$ messages.

For the 2-d decomposition, for each node we need to send the first and last row to the top and bottom neighbors, first and last columns to left and right neighbors. Additionally we need to send the four corners to their corresponding neighbors. So, for each node we need to send out 8 different messages. Thus, for the decomposition we should send $8 \times p$ messages. Where p is the number of processors or nodes.

What is the total amount of data sent by a node in each decomposition? Express your answer in number of cells/words transmitted (not bytes).
Answer:

For horizontal decomposition, each node must send $2 \times n$ cells. For the 2-d case a node as we discussed we need to send the borders to the 8 neighbors. Therefore, we need to send 4 messages of length $\frac{n}{\sqrt{p}}$ and 4 messages of length 1 which are the corners. So the total number cells we to be transmitted is $4 \times \frac{n}{\sqrt{p}} + 4$.

Q2- The answers to question 1 define a tradeoff in which you would choose different messaging disciplines:

1- When messages are small and latency dominates messaging performance which decomposition would you prefer and why?
Answer:
When latency dominates we need to use less number of messages because for each message we have a latency and we want to minimize it. Therefore, horizontal decomposition is more efficient for this case as it has less number of transfer messages ($2p < 8p$).

2- When messages are large and throughput dominates messaging performance which decomposition would you prefer and why?
Answer:
When throughput dominates, we should try to send smaller messages. Consequently, we prefer the 2-d decomposition as it has smaller messages ($\frac{n}{\sqrt{p}} < n$). Also it has less number of data to be send as we saw in previous part. Because $4 \times \frac{n}{\sqrt{p}} + 4 < 2n$ if p is large enough.

Q3. For the 2-d decomposition, design a deadlock-free messaging discipline for transmitting the top, bottom, left, and right sides and the top left, top right, bottom left and bottom right corners among neighboring partitions. Draw and submit pictures of the messages sent and received by a single node. Each drawing will describe one of multiple rounds, akin to the figure in Lecture 16. In each round a node may either send or receive from multiple neighbors. Note: We understand that not all nodes conduct the same protocol. Draw pictures from the perspective of any node.

Answer:
We can have several ways to do that. The messaging should be in a way that a node doesn't receive and send data to a neighbor simultaneously. An example is shown in figures 1 and 2. Read arrow shows sending and green arrow shows receiving.

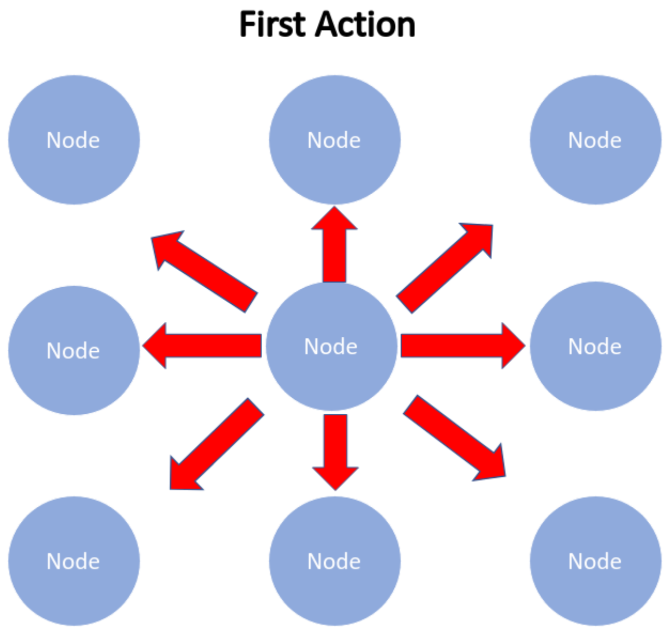

Figure 1: First Action

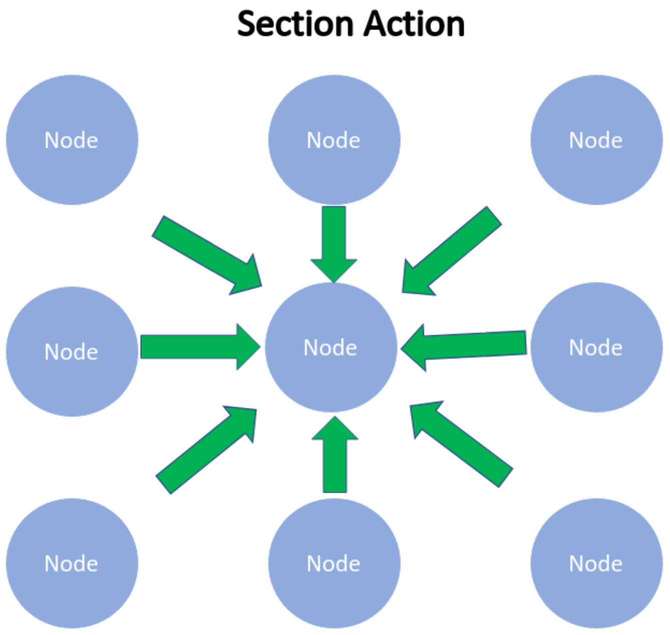

Figure 2: Second Action