Bardia Mojra – 1000766739

07/24/2022 – Assignment 04

Q1. This is a shared memory C implementation.

typedef struct {

double value;

int outNode;

} ScanNodePair;

void c\_Scan(ScanNodePair \*inNode, ScanNodePair \*inoutNode)

{

int i;

ScanNodePair scan;

for (i = 0; i < 8; ++i) {

if (inNode->outNode == inoutNode->outNode)

scan.value = inNode->value + inoutNode->value;

else

scan.value = inoutNode->value;

scan.outNode = inout->outNode;

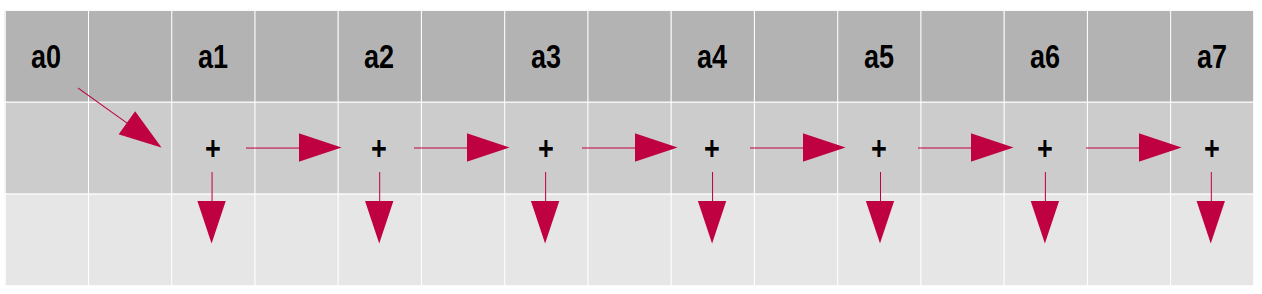
\*inout = scan;

inNode++;

inoutNode++;

}

}



Q2.

Row-wise partitioning efficiency:

On a hypercube communication network, the row-wise algorithm has an overall communication time of , where is the communication startup time and is the per word communication time. The efficiency is E = , where is the communication overhead, n is the dimension length of the matrix, p is the number of processors, and is unit computation time. The isoefficiency is obtained by . For checkerboard partitioning, parallel time is approximated by , so isoefficiency is as it is the dominant term as p gets larger. Given the two isoefficiency, we conclude the checkerboard algorithm is more stable as its time complexity grows at a much lower rate as you increase the processor count. Nonetheless, it is important to note that this only applies to a well-balanced hypercube parallel computing system. For a hypercube communication topology, the overall isoefficiency is heavily dependent on the ratio of per-word communication time and compute time as defined by . The lower E ratio (generally less than .5) results in higher isoefficiency rates, given enough communication bandwidth by hardware.

Q3. Time for sequential process is given by and a parallel time of .

1. Fixed-workload speedup analysis is based on Ahmdal’s law,

speedup =

Thus, we can factor out W (fixed workload) and efficiency becomes purely dependant on communication time . For very large processor count, speed up approaches 1/where is between 0 and 1.

1. Fixed-time speedup is based on Gustafson's law which assumes a fixed execution time for all processes. Now, we have,

speedup =

Which becomes which indicates that the speedup increases as workload increases for an increased number of processors with a fixed computation time.

Q4. E=.25, B is more scalable than A. b,c>0 and , and

1. ,
2. ,

Since the iso-efficiency functions are given, we obtain B scalability factor over A by,

, Since lower W implies higher scalability factor, we conclude B scalability factor over A is .

,

For E=¼:

For A,

And for B,

Now if we consider b=4c, we will have,

Thus, , under the given condition, system B is more scalable and to maintain the same isoefficiency the workload has to increase by a factor of (2.82)^(-1), which is .

Yes, B becomes slower if E increases to .5 and beyond.

Q5.

1. 1.71
2. 4
3. 10
4. 1.93
5. 34
6. 100