EE451 Homework3

JunqiHu 4101333969

18 de septiembre de 2020

Cont

1. **Problem 1**

   a) *PRAM : A model consists of p processors and a global memory of unbounded size that is uniformly accessible to all processors. All processors access the same address space and share a common clock but may execute different instructions in each cycle.*
   b) *Cannon′s algorithm : First we select the k in every iteration for every processor element to access the data aik ∗ bkj, Then we distribute the input matrices between the processor based on previous rule then we cyclically update the a an*
   c) *Blocking and non − blocking send/receive : blocking Message Passing Operations is for the send operation to return onl blocking protocols returns from the send or receive operation before it is seantically safe to do so.*
   d) *Single Program Multiple Data : Itmeanswerunthesameprogramonmultipleprocessorplatformwithdifferentdata*
   e) *Loosely synchronous : tasks or subsets of tasks synchronize to perfor interactions.However, between these interactions,*

2. *Problem* 2
   I think using the openmp this way could not obtain the same result
   as the serial execution in some situation.
   Every computing instruction depends on the left side and up side element
   while those element is the result of other threads.
   It means thatthe result would varies considerating the speed of different threads.

3. *Problem* 3
   I think using the openmp this way could not obtain the same result
   as the serial execution in some situation.Every computing instruction depends on the left side and up side element
   while those element is the result of other threads.
   It means that the result would varies considerating the speed of different threads.

4. **Problem 4**

   a) *If using blockingsend/receive, since in the INS3 process A and process B are executing the send,*
   they would wait for other process to end.
   However there isn′t other process, so there would be a deadlock.
   To avoid deadlock, I would change the line 3 in process B to "send(b, 1, A)"
   b) *It is not safe to immediately execute because both line3 INS and line4 INS are modifying the variable A.*
   They might not get the expected result given the running speed.

5. **Problem 5**

   a)
   ```
   $(processi, j)$
   INIT GlobalVariableC
   INIT LocalVariable : A B
   SUPER STEP :
   $C[i][j] = A[i][k] * B[k][j]$
   $SEND(A, 1, thread[i-1][j])$
   $RECEIVE(B, 1, thread[i][j+1])$
   ```

   b) *It is not safe to immediately execute because both line3 INS and line4 INS are modifying the variable A.*
   They might not get the expected result given the running speed.