CS4823/CS6643 Parallel Computing
Fall, 2020

Test 1

Time = 75 mins. Attempt all questions – 3(b) is bonus for undergrads. Show your work, as grading would be done based on your approach and steps taken, as well as on correctness. Present all algorithms at an abstract level (i.e. give pseudo codes only if absolutely necessary, or when asked). Be neat and precise. Partial marks will be given. Open book and notes.

1. *Architectures:*

   (a) (3 points) Distinguish between SIMD and MIMD computers. Is Fox server as SIMD machine?

   (b ~~(a)~~ (2+3 points) (i) Draw a 4-dimensional hypercube network and show its binary labels. (ii) Give a self-routing scheme to send a message from processor Pi to Pj in O(*log p*) time, if *p* is number of processors (graduate students: prove the time complexity).

   (c) ~~(b)~~ (4 points) Estimate, with 1-2 sentence reasoning, the diameter, max degree, number of edges, and bisection width of a fat tree of $n = 2^k$-1 nodes using order notation.

2. *Algorithms and Analysis:* Given *n* binary bits in an array A[0..n-1] in the shared memory. Present a PRAM algorithm to find XOR of bits in array A, and analyze it. For this,

   (a) (4 points) Using *p* processors, for *p* up to *n*, state the basic steps of your algorithm (3-4 sentences).

   (b) (3 + 1 points) Calculate parallel time T_p and work done. Give expressions for speedup S_p and the cost of this algorithm.

   (c) (3 points) Show that this algorithm achieves cost optimality (i.e., O(p) speedup) for processors in the range 1 through *O*(n/log n).

   (d) (1 point) State the version of PRAM you are using, and define it.

3. *Programming:*

   (a) *(6 points)* Given *n* binary bits in arrays A[0..n-1] and B[0..n-1] in the shared memory. *S*ketch an efficient OpenMP program for finding scalar product of array A and B by giving either pseudo-codes or C codes (i.e., multiply corresponding bits of A and B, and then add them). Assume the answer should be in a shared variable *scalar_product*.

   (b) (5 points) (Bonus for undergrads) If, instead of adding the n product bits, suppose you had to find Boolean OR of the product bits in a shared variable *scalar_OR*. How would you use the idea of the constant time algorithm for Boolean OR on CRCW PRAM to do this without employing a reduction tree? Calculate parallel time T_p for the algorithms in (a) and (b). Show that algorithm in (b) achieves better range of cost optimality.