- Middle East Technical University

Department of Computer Engineering

CENG 478

Introduction to Parallel Computing

Spring 2019-2020

Assignment 4

Due date: 03.06.2020, 23:59

1. Questions
   a. (30 pts) Consider another load-balancing strategy. Assume that each processor maintains a variable called counter. Initially, each processor initializes its local copy of counter to zero. Whenever a processor goes idle, it searches for two processors P_i and P_{i+1} in a logical ring embedded into any architecture, such that the value of counter at P_i is greater than that at P_{i+1}. The idle processor then sends a work request to processor P_{i+1}. If no such pair of processors exists, the request is sent to processor zero. On receiving a work request, a processor increments its local value of counter.

      Devise algorithms to detect the pairs P_i and P_{i+1}. Analyze the scalability of this loadbalancing scheme based on your algorithm to detect the pairs P_i and P_{i+1} for a message passing architecture.

      Hint: The upper bound on the number of work transfers for this scheme is similar to that for GRR. (Problem 11.2)

   b. (30 pts) Consider a parallel formulation of best-first search of a graph that uses a hash function to distribute nodes to processors (Section 11.5). The performance of this scheme is influenced by two factors: the communication cost and the number of "good" nodes expanded (a "good" node is one that would also be expanded by the sequential algorithm). These two factors can be analyzed independently of each other.

      Assuming a completely random hash function (one in which each node has a probability of being hashed to a processor equal to 1/p), show that the expected number of nodes expanded by this parallel formulation differs from the optimal number by a constant factor (that is, independent of p). Assuming that the cost of communicating a node from one processor to another is O (1), derive the isoefficiency function of this scheme. (Problem 11.10)

   c. (40 pts) Let the serial run time of an n-point FFT computation be $t_c n \log n$. Consider its implementation on an architecture on which the parallel run time is $(t_c n \log n)/p + (t_w n \log p)/p$. Assume that $t_c = 1$ and $t_w = 0.2$. (Problem 13.1)

      c.1. Write expressions for the speedup and efficiency.
      c.2. What is the isoefficiency function if an efficiency of 0.6 is desired?
      c.3. How will the isoefficiency function change (if at all) if an efficiency of 0.4 is desired?
      c.4. Repeat parts 1 and 2 for the case in which $t_w = 1$ and everything else is the same.

2. Notes

   a. You will submit a pdf or jpeg file via ODTÜClass. It is not important whether you prepare your answers on a digital platform or just on a paper. Just be sure that you have fully **explained** your answers and the photo taken is **readable**.

   b. You can still submit your work if the **deadline** is passed, however with an increasing **penalty** of **5*days*days**. (i.e. first day -5 points, second day -5*2*2=-20 points and so on). Note that even a minute late means that it is the other day.

   c. We have zero tolerance policy for cheating. People involved in cheating will be punished according to the university regulations and will get zero.