High Performance Programming [COMTEK3, & ESD3]
Lecture 1: Introduction to Parallel Processing 02/2025

Instruction: *The exercise problems are designed to aid your understanding of the concepts covered during the lecture. They are intended to be discussed and solved in a group of* $3-6$ *students. The most important part of the exercise is the* ***learning process****. It is strongly recommended that you avoid the usage of LLMs such as chatGPT and similar AI models to generate solutions* ***without trying to solve the problems first****.*

Exercise 1: **Parallel Computing Models and Scheduling**

Sketch the binary tree models and specify a possible schedule for the summation of 16 numbers - $a_1, a_2, \cdots, a_{16}$ using

(a) a single processor (sequential computation)

(b) 4 processors

(c) 5 processors

(d) 8 processors

(e) 16 processors

(f) Discuss the potential speedup from your schedule in b - e.

(g) Determine the efficiency of your schedule in b - e.

Exercise 2: **Parallel Computing Models and Performance**

Consider the problem of finding the value of the polynomial:

$$f(x) = \sum_{i=0}^{10} a_i x^i$$

(a) Assuming that ten processors are available to work in parallel, draw the binary tree model of the computation.

(b) What is the number of processors for achieving maximum speedup and efficiency?

Exercise 3: **Parallel Processing Effectiveness**

An image processing application problem is characterized by 12 unit-time tasks:

1. an input task that must be completed before any other task can start and consumes the entire bandwidth of the single-input device available,
2. 10 completely independent computational tasks, and
3. an output task that must follow the completion of all other tasks and consumes the entire bandwidth of the single-output device available.

Assume the availability of one input and one output device throughout.

(a) What is the maximum speed-up that can be achieved for this application with two processors?

(b) What is an upper bound on the speed-up with parallel processing?

(c) How many processors are sufficient to achieve the maximum speed-up derived in part (b)?

(d) What is the maximum speed-up in solving five independent instances of the problem on two processors?

(e) What is an upper bound on the speed-up in parallel solution of 100 independent instances of the problem?

(f) How many processors are sufficient to achieve the maximum speed-up derived in part (f)?

(g) What is an upper bound on the speed-up, given a steady stream of independent problem instances?

Exercise 4: **Amdahl's Law**

Amdahl's law can be applied in contexts other than parallel processing. Suppose that a numerical application consists of 20% floating-point and 80% integer/control operations (these are based on operation counts rather than their execution times). The execution time of a floating-point operation is three times as long as other operations. We are considering a redesign of the floating-point unit in a microprocessor to make it faster.

(a) Formulate a more general version of Amdahl's law in terms of selective speed-up of a portion of a computation rather than in terms of parallel processing.

(b) How much faster should the new floating-point unit be for 25% overall speed improvement?

(c) What is the maximum speed-up that we can hope to achieve by only modifying the floating-point unit?

RAD