FIT3143 Tutorial Week 2

Lecturers: ABM Russel (MU Australia) and Vishnu Monn (MU Malaysia)

INTRODUCTION

OBJECTIVES

- The purpose of this tutorial is to introduce Parallel Computing
- Understand theoretical performance

Note: Tutorials are not assessed. Nevertheless, please attempt the questions to improve your unit comprehension in preparation for the labs, assignments, and final assessments.

QUESTIONS/TOPICS

SAMPLE SOLUTIONS

1. Briefly discuss the advantages of using Parallel computing.

   Speed, complex problem solving etc. Students may discuss any discipline specific examples.

2. In theory, the maximum aggregated performance of a single compute node can be measured in terms of Maximum aggregate floating-point operations using the formula:

   $$P = N \times C \times F \times R$$

   Where P performance in flops, N number of nodes, C number of CPUs, F floating point ops per clock period - FLOP, R clock rate. Using this formula, estimate the maximum theoretical dual precision performance of your personal laptop (or desktop computer). HINT: FLOPS.

   P = N ×C ×F ×R If my laptop is running a 2.3 GHz (R = 2.3 × 109) Intel i7, Haswell generation, giving me 16 dual precision ops per cycle (F = 16). I have a single laptop (N = 1), with a single CPU (C = 1). Putting these together my machine should be capable of P = 1×1×16×2.3×10^9 = 36.8 GFLOPS.

3. 20% of a program's execution time is spent within inherently sequential code. What is the limit to the speedup achievable by a parallel version of the program?

HINT: Amdahl's law as follows:

$$y \leq \frac{1}{f + (1 - f)/p}$$

Where,

f = fraction of the code that is inherently sequential (also known as r_s)

$1\text{-}f$ = fraction of the code that can be parallelized (also known as r_p)

p = number of processors

Assume p is infinity then 1/(.2) = 5
Max speed-up with infinite processors cannot be more than 5.

4. A computer animation program generates a feature movie frame-by-frame. Each frame can be generated independently, and the generated frame is output serially to a file. It takes 99 seconds to render a frame and 1 second to output it. The rendering process of a single frame can be parallelized but writing the frame to file is done serially. If we intend to parallelize the process of rendering a single frame, how much theoretical speedup can be achieved using 100 processors?

   For a single frame, rendering = 99 seconds, writing to file = 1 second. Total time to render and write a single frame = 100 seconds.

   $r_s = 1/100 = 0.01$

   $r_p = 1 – rs = 0.99$

   $p = 100$

   $S_{theory}(p) = 1/ (r_s + r_p /p))$

   $S_{theory}(100) = 1 / (0.01 + 0.99/100)) = 50.25$

5. Discuss the limitations of Amdahl's law.

   - Ignores k(n, p) - overestimates speedup. Kappa, k is the communication time.
   - Assumes *f* constant, so underestimates speedup achievable.

6. A parallel program executing on 32 processors spends 5% of its time in sequential code. What is the scaled speedup of this program?

   $S_{theory}(p) = p + (1 – p)r_s$

   $S_{theory}(32) = 32 + (1\text{-}32)0.05 = 30.45$