CENG3420 Homework 4

NO need to submit

Solutions

Q1 (25%) This question is about Amdahl's Law:

$$\text{Speedup due to enhancement } E = \frac{1}{(1-F)+F/S}$$

where F is the fraction that can get speedup, while S is the speedup factor.

1. Consider an enhancement which runs 20 times faster but which is only usable 25% of the time. Calculate E.
2. Consider summing 100 scalar variables and two 10×10 matrices (matrix sum) on 10 processors. Calculate E.

A1

1. According to the question $F = 0.25, S = 20$, substitute to the speedup equation,

$$E = \frac{1}{1-0.25+0.25/20} = 1.31. \tag{1}$$

2. There are total 200 operations, where scalar operations are not parallelizable and matrix addition is parallelizable. Here $F = 0.5, S = 10$, and

$$E = \frac{1}{1-0.5+0.5/10} = 1.82. \tag{2}$$

Q2 (**25%**) In the design of a multi-core processor, there are fixed on chip cache resources. We assume maximum of n cores can be designed with those resources. Let k be the real designed core number ($r = \frac{n}{k}$ is integer.) Define a speed up factor $s(r)$ as sequential performance gain by using the resources equivalent to r cores to form a single core, and obviously $s(1) = 1$. Given f the fraction of software that is parallelizable across multiple cores, prove the speed up of the multi-core processor in terms of f, r, n, and $s(r)$ is

$$S(f,r,n) = \frac{1}{\frac{1-f}{s(r)} + \frac{f \times r}{n \times s(r)}} \tag{3}$$

A2

$$\begin{aligned} S(f,r,n) &= s(r) \times \frac{1}{(1-f)+\frac{f}{k}} \\ &= s(r) \times \frac{1}{(1-f)+\frac{f\times r}{n}} \\ &= \frac{1}{\frac{1-f}{s(r)} + \frac{f \times r}{n \times s(r)}}. \end{aligned} \tag{4}$$

Q3 (**25%**) Consider the following portions of two different programs running at the same time on four processors in a share memory multiprocessor (SMP). Assume that before this code is run, both x and y are 0.

Core1: `x = 3;`

Core2: `y = 3;`

Core3: `w = x + y + 1;`

Core4: `z = x - y;`

Core5: `r = w + z;`

1. What are all the possible resulting values of w, x, y, z, and r? For each possible outcome, explain how we might arrive at those values.

A3 1. As shown in the following table:

Table 1: One correct column for 1 point

x	3	3	3	3	3	3	3	3	3
y	3	3	3	3	3	3	3	3	3
w	1	1	1	4	4	4	7	7	7
z	0	-3	3	0	-3	3	0	-3	3
r	1	-2	4	4	1	7	7	4	10

Q4 (**25%**) Given an original code as follows:

```
Loop:  L.D     F0,0 (R1)      ; F0=array element
       ADD.D   F4, F0, F2     ; add scalar in F2
       S.D     F4, 0 (R1)     ; store result
       DADDUI  R1, R1, #-8    ; decrement pointer 8 bytes
       BNE     R1, R2, Loop   ; branch R1!=R2
```

1. Please revise the original code to the code with loop unrolling (4 times).
2. Based on the revised the code with loop unrolling, please revise the code with pipeline scheduling.

A4 (1)

```
Loop:
    L.D     F0, 0 (R1)
    ADD.D   F4, F0, F2
    S.D     F4, 0 (R1)     ; drop DADUI & BNE
    L.D     F6, -8 (R1)
    ADD.D   F8, F6, F2
    S.D     F8, -8 (R1)    ; drop DADDUI & BNE
    L.D     F10, -16 (R1)
    ADD.D   F12, F10, F2
    S.D     F12, -16 (R1) ; drop DADDUI & BNE
```

```
    L.D     F14, -24 (R1)
    ADD.D   F16, F14, F2
    S.D     F16, -24 (R1)
    DADDUI  R1, R1, #-32
    BNE     R1, R2, Loop
```

(2)

```
Loop:
    L.D     F0, 0 (R1)
    L.D     F6, -8 (R1)
    L.D     F10, -16 (R1)
    L.D     F14, -24 (R1)
    ADD.D   F4, F0, F2
    ADD.D   F8, F6, F2
    ADD.D   F12, F10, F2
    ADD.D   F16, F14, F2
    S.D     F4, 0 (R1)
    S.D     F8, -8 (R1)
    DADDUI  R1, R1, #-32
    S.D     F12, 16 (R1)
    BNE     R1, R2, Loop
    S.D     F16, 8 (R1); 8-32 = -24
```