CPSC 479 Homework 1: Introduction to HPC

Prof. Doina Bein, CSU Fullerton

dbein@fullerton.edu

Submission: Upload the answers on Canvas. In case of uploading multiple files, clearly labeling each file.

Problem 1. [10 points] Consider the following instructions of a sequential program:
I1 : z = (b * d) / (b + d)
I2 : x = z*z + (c * a)
I3 : y = (e + a) * d
I4: z = x + d / (b + c)

(a) Which pairs of instructions (I1-I2, I1-I3, I1-I4, I2-I3, I2-I4, I3-I4) can be parallelized? Use the data dependency conditions to decide which instructions can be executed in parallel. Show your complete work otherwise points will be deducted. [1.5 points each]
(b) Can all four I1-I2-I3-I4 be parallelized? Justify a NO or YES answer. [1 point]

Show your work.

Answer:

I1: in = {b, d} out = {z}

I2: in = {z, c, a} out = {x}

I3: in = {e, a, d} out = {y}

I4: in = {x, d, b, c} out = {z}

(a) 1. I1 || I2? No

f.d: out(1) ∩ in(2) = {z} ∩ {z, c, a} = {z}
a.d: in(1) ∩ out(2) = {b,d} ∩ {x} = Ø
o.d: out(1) ∩ out(2) = {z} ∩ {x} = Ø

2. I1 || I3? Yes

f.d: out(1) ∩ in(3) = {z} ∩ {e, a, d} = Ø
a.d: in(1) ∩ out(3) = {b, d} ∩ {y} = Ø
o.d: out(1) ∩ out(3) = {z} ∩ {y} = Ø

3. I1 || I4? No

f.d: out(1) ∩ in(4) = {z} ∩ {x, d, b, c} = Ø
a.d: in(1) ∩ out(4) = {b, d} ∩ {z} = Ø
o.d: out(1) ∩ out(4) = {z} ∩ {z} = {z}

4. I2 || I3? Yes

f.d: out(2) ∩ in(3) = {x} ∩ {e, a, d} = Ø
a.d: in(2) ∩ out(3) = {z, c, a} ∩ {y} = Ø

o.d: $out(2) \cap out(3) = \{x\} \cap \{y\} = \emptyset$

5. I2 || I4? No

f.d: $out(2) \cap in(4) = \{x\} \cap \{x, d, b, c\} = \{x\}$
a.d: $in(2) \cap out(4) = \{z, c, a\} \cap \{z\} = \{z\}$
o.d: $out(2) \cap out(4) = \{x\} \cap \{z\} = \emptyset$

6. I3 || I4? Yes

f.d: $out(3) \cap in(4) = \{y\} \cap \{x, d, b, c\} = \emptyset$
a.d: $in(3) \cap out(4) = \{e, a, d\} \cap \{z\} = \emptyset$
o.d: $out(3) \cap out(4) = \{y\} \cap \{z\} = \emptyset$

(b) All four cannot be parallelized. Instructions I1 and I2 have a flow dependency, I1 and I4 have an output dependency, and I2 and I4 have flow and anti dependencies.

Problem 2. [2 points] Consider the following sequential program:

x[0] = 1
x[1] = x[0] + 2
x[2] = x[1] + 3
x[3] = 3
x[4] = x[2] + 3
x[5] = x[1] + 2

The fastest execution time of this portion of the program can be obtained by executing this code in parallel. What is the number of processors that can be used to execute this code in parallel and obtain the minimum execution time? You do not need to use Amdahl's law. Show your complete work to justify your answer otherwise points will be deducted. [2 points]

Answer:

I1: In = Ø Out = {x[0]}

I2: In = {x[0]} Out = {x[1]}

I3: In = {x[1]} Out = {x[2]}

I4: In = Ø Out = {x[3]}

I5: In = {x[2]} Out = {x[4]}

I6: In = {x[1]} Out = {x[5]}

1. I1 || I2? NO

f.d: $out(1) \cap in(2) = \{x[0]\} \cap \{x[0]\} = \{x[0]\}$
a.d: $in(1) \cap out(2) = \emptyset \cap \{x[1]\} = \emptyset$
o.d: $out(1) \cap out(2) = \{x[0]\} \cap \{x[1]\} = \emptyset$

2. I1 || I3? YES

f.d: $out(1) \cap in(3) = \{x[0]\} \cap \{x[1]\} = \emptyset$
a.d: $in(1) \cap out(3) = \emptyset \cap \{x[2]\} = \emptyset$
o.d: $out(1) \cap out(3) = \{x[0]\} \cap \{x[2]\} = \emptyset$

3. I1 || I4? YES

f.d: $out(1) \cap in(4) = \{x[0]\} \cap \emptyset = \emptyset$
a.d: $in(1) \cap out(4) = \emptyset \cap \{x[3]\} = \emptyset$
o.d: $out(1) \cap out(4) = \{x[0]\} \cap \{x[3]\} = \emptyset$

4. I1 || I5? YES

f.d: $out(1) \cap in(5) = \{x[0]\} \cap \{x[2]\} = \emptyset$
a.d: $in(1) \cap out(5) = \emptyset \cap \{x[4]\} = \emptyset$
o.d: $out(1) \cap out(5) = \{x[0]\} \cap \{x[4]\} = \emptyset$

5. I1 || I6? YES

f.d: $out(1) \cap in(6) = \{x[0]\} \cap \{x[1]\} = \emptyset$
a.d: $in(1) \cap out(6) = \emptyset \cap \{x[5]\} = \emptyset$
o.d: $out(1) \cap out(6) = \{x[0]\} \cap \{x[5]\} = \emptyset$

6. I2 || I3? NO

f.d: $out(2) \cap in(3) = \{x[1]\} \cap \{x[1]\} =\{x[1]\}$
a.d: $in(2) \cap out(3) = \{x[0]\} \cap \{x[2]\} = \emptyset$
o.d: $out(2) \cap out(3) = \{x[1]\} \cap \{x[2]\} = \emptyset$

7. I2 || I4? YES

f.d: $out(2) \cap in(4) = \{x[1]\} \cap \emptyset = \emptyset$
a.d: $in(2) \cap out(4) = \{x[0]\} \cap \{x[3]\} = \emptyset$
o.d: $out(2) \cap out(4) = \{x[1]\} \cap \{x[3]\} = \emptyset$

8. I2 || I5? YES

f.d:	$out(2) \cap in(5) = \{x[1]\} \cap \{x[2]\} = \emptyset$
a.d:	$in(2) \cap out(5) = \{x[0]\} \cap \{x[4]\} = \emptyset$
o.d:	$out(2) \cap out(5) = \{x[1]\} \cap \{x[4]\} = \emptyset$

9. I2 || I6? NO

f.d:	$out(2) \cap in(6) = \{x[1]\} \cap \{x[1]\} = \{x[1]\}$
a.d:	$in(2) \cap out(6) = \{x[0]\} \cap \{x[5]\} = \emptyset$
o.d:	$out(2) \cap out(6) = \{x[1]\} \cap \{x[5]\} = \emptyset$

10. I3 || I4? YES

f.d:	$out(3) \cap in(4) = \{x[2]\} \cap \emptyset = \emptyset$
a.d:	$in(3) \cap out(4) = \{x[1]\} \cap \{x[3]\} = \emptyset$
o.d:	$out(3) \cap out(4) = \{x[2]\} \cap \{x[3]\} = \emptyset$

11. I3 || I5? NO

f.d:	$out(3) \cap in(5) = \{x[2]\} \cap \{x[2]\} = \{x[2]\}$
a.d:	$in(3) \cap out(5) = \{x[1]\} \cap \{x[4]\} = \emptyset$
o.d:	$out(3) \cap out(5) = \{x[2]\} \cap \{x[4]\} = \emptyset$

12. I3 || I6? YES

f.d:	$out(3) \cap in(6) = \{x[2]\} \cap \{x[1]\} = \emptyset$
a.d:	$in(3) \cap out(6) = \{x[1]\} \cap \{x[5]\} = \emptyset$
o.d:	$out(3) \cap out(6) = \{x[2]\} \cap \{x[5]\} = \emptyset$

13. I4 || I5? YES

f.d:	$out(4) \cap in(5) = \{x[3]\} \cap \{x[2]\} = \emptyset$
a.d:	$in(4) \cap out(5) = \emptyset \cap \{x[4]\} = \emptyset$
o.d:	$out(4) \cap out(5) = \{x[3]\} \cap \{x[4]\} = \emptyset$

14. I4 || I6? YES

f.d:	$out(4) \cap in(6) = \{x[3]\} \cap \{x[1]\} = \emptyset$
a.d:	$in(4) \cap out(6) = \emptyset \cap \{x[5]\} = \emptyset$
o.d:	$out(4) \cap out(6) = \{x[3]\} \cap \{x[5]\} = \emptyset$

15. I5 || I6? YES

f.d:	out(5) ∩ in(6) = {x[4]} ∩ {x[1]} = Ø
a.d:	in(5) ∩ out(6) = {x[2]} ∩ {x[5]} = Ø
o.d:	out(5) ∩ out(6) = {x[4]} ∩ {x[5]} = Ø

Schedule:

Processor:	1	2	3	4
	I1	I3	I6	I4

	I2	I5		

Four processors can be used to achieve the minimum execution time.

Problem 3.[2 points] Amdahl's law computes the speedup in execution time for a program executed on P processors with fixed input size. If the input size increases proportionally to the number of processors, then Gustafson law is used to calculate the speedup. Give two real-world examples of problems and/or situations in which Gustafson law should be applied instead of Amdahl's law. Justify your answer otherwise points will be deducted. [2 points]

Answer: Two examples are video processing and medical imaging, where the resolution and thus the size of the data will increase given the computing resources available (Nielsen 2016)

References

Nielsen, F. (2016). *Introduction to HPC with MPI for data science*. Springer.

.