|  |
| --- |
|  |
| **Assignment # 1**  **SYSC 5207 – Distributed Systems Engineering** |
| Fall 2014  Submitted To  Dr. Shikharesh Majumdar  By  **Ferhan Jamal (100953487)**  **Rudraneel chakraborty (7563772, 100992549)** |

**Answer 1:**

a. Complete specification (list of processes and partial order) for π =

P= {P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11)

<. ={(P1, P3), (P1, P4), (P2, P5), (P2, P6), (P4, P7), (P5, P8), (P5, P9), (P6, P8), (P7, P9), (P9, P10), (P9, P11)}

No, the computation is not determinate. The **connectivity between node 3 and node 2** and the **connectivity between node 3 and node 7** are required to make the computation determinate. The explanation of that is explained below:

We have to consider the various combinations of nodes in the precedence graph to prove that the **connectivity between node 3 and node 2** and the **connectivity node 3 and node 7** will make the computation determinate. The various combinations are as follows:-

1, 2 2, 1 3, 1 4, 1 5, 1

1, 3 2, 3 3, 2 4, 2 5, 2

1, 4 2, 4 3, 4 4, 3 5, 3

1, 5 2, 5 3, 5 4, 5 5, 4

1, 6 2, 6 3, 6 4, 6 5, 6

1, 7 2, 7 3, 7 4, 7 5, 7

1, 8 2, 8 3, 8 4, 8 5, 8

1, 9 2, 9 3, 9 4, 9 5, 9

1, 10 2, 10 3, 10 4, 10 5, 10

1, 11 2, 11 3, 11 4, 11 5, 11

­

6, 1 7, 1 8, 1 9, 1 10, 1 11, 1

6, 2 7, 2 8, 2 9, 2 10, 2 11, 2

6, 3 7, 3 8, 3 9, 3 10, 3 11, 3

6, 4 7, 4 8, 4 9, 4 10, 4 11, 4

6, 5 7, 5 8, 5 9, 5 10, 5 11, 5

6, 7 7, 6 8, 6 9, 6 10, 6 11, 6

6, 8 7, 8 8, 7 9, 7 10, 7 11, 7

6, 9 7, 9 8, 9 9, 8 10, 8 11, 8

6, 10 7, 10 8, 10 9, 10 10, 9 11, 9

6, 11 7, 11 8, 11 9, 11 10, 11 11, 10

In the various combinations as described above, 1st we will remove that combinations that are connected with link and 2nd we will remove the ones that have no memory problem. We are doing so because in this way we can find the bad cases.

**a. Firstly, we are removing the ones that are connected with link including both direct and indirect predecessors and successors relations.**

1, 2 2, 1 ~~3, 1~~ ~~4, 1~~ 5, 1

~~1, 3~~ 2, 3 3, 2 4, 2 ~~5, 2~~

~~1, 4~~ 2, 4 3, 4 4, 3 5, 3

1, 5 ~~2, 5~~ 3, 5 4, 5 5, 4

1, 6 ~~2, 6~~ 3, 6 4, 6 5, 6

~~1, 7~~ 2, 7 3, 7 ~~4, 7~~ 5, 7

1, 8 ~~2, 8~~ 3, 8 4, 8 ~~5, 8~~

~~1, 9~~ ~~2, 9~~ 3, 9 ~~4, 9~~ ~~5, 9~~

~~1, 10~~ ~~2, 10~~ 3, 10 ~~4, 10~~ ~~5, 10~~

~~1, 11~~ ~~2, 11~~ 3, 11 ~~4, 11~~ ~~5, 11~~

­

6, 1 ~~7, 1~~ 8, 1 ~~9, 1~~ ~~10, 1~~ ~~11, 1~~

~~6, 2~~ 7, 2 ~~8, 2~~ ~~9, 2~~ ~~10, 2~~ ~~11, 2~~

6, 3 7, 3 8, 3 9, 3 10, 3 11, 3

6, 4 ~~7, 4~~ 8, 4 ~~9, 4~~ ~~10, 4~~ ~~11, 4~~

6, 5 7, 5 ~~8, 5~~ ~~9, 5~~ ~~10, 5~~ ~~11, 5~~

6, 7 7, 6 ~~8, 6~~ 9, 6 10, 6 11, 6

~~6, 8~~ 7, 8 8, 7 ~~9, 7~~ ~~10, 7~~ ~~11, 7~~

6, 9 ~~7, 9~~ 8, 9 9, 8 10, 8 11, 8

6, 10 ~~7, 10~~ 8, 10 ~~9, 10~~ ~~10, 9~~ ~~11, 9~~

6, 11 ~~7, 11~~ 8, 11 ~~9, 11~~ 10, 11 11, 10

**b. Secondly, we are removing the ones that have no memory problem from the case or combinations which have no links highlighted by yellow.**

1, 2 2, 1 ~~3, 1~~ ~~4, 1~~ 5, 1

~~1, 3~~ 2, 3 3, 2 4, 2 ~~5, 2~~

~~1, 4~~ 2, 4 3, 4 4, 3 5, 3

1, 5 ~~2, 5~~ 3, 5 4, 5 5, 4

1, 6 ~~2, 6~~ 3, 6 4, 6 5, 6

~~1, 7~~ 2, 7 3, 7 ~~4, 7~~ 5, 7

1, 8 ~~2, 8~~ 3, 8 4, 8 ~~5, 8~~

~~1, 9~~ ~~2, 9~~ 3, 9 ~~4, 9~~ ~~5, 9~~

~~1, 10~~ ~~2, 10~~ 3, 10 ~~4, 10~~ ~~5, 10~~

~~1, 11~~ ~~2, 11~~ 3, 11 ~~4, 11~~ ~~5, 11~~

­

6, 1 ~~7, 1~~ 8, 1 ~~9, 1~~ ~~10, 1~~ ~~11, 1~~

~~6, 2~~ 7, 2 ~~8, 2~~ ~~9, 2~~ ~~10, 2~~ ~~11, 2~~

6, 3 7, 3 8, 3 9, 3 10, 3 11, 3

6, 4 ~~7, 4~~ 8, 4 ~~9, 4~~ ~~10, 4~~ ~~11, 4~~

6, 5 7, 5 ~~8, 5~~ ~~9, 5~~ ~~10, 5~~ ~~11, 5~~

6, 7 7, 6 ~~8, 6~~ 9, 6 10, 6 11, 6

~~6, 8~~ 7, 8 8, 7 ~~9, 7~~ ~~10, 7~~ ~~11, 7~~

6, 9 ~~7, 9~~ 8, 9 9, 8 10, 8 11, 8

6, 10 ~~7, 10~~ 8, 10 ~~9, 10~~ ~~10, 9~~ ~~11, 9~~

6, 11 ~~7, 11~~ 8, 11 ~~9, 11~~ 10, 11 11, 10

The combinations which are in blue color does not satisfy Bernstein conditions and are not in predecessors and successors relationship so the computation is not determinate. The additions of link **between node 3 and node 2** and between node **3 and node 7** in the precedence graph below will make the computation determinate.

**P2**

**P1**

**P3 P4 P5 P6**

**P7 P8**

**P9**

**P10**

**P11**

**Answer 2:**

The precedence graph mentioned in question 2 for the parbegin-parend construct are as follows:

Model for the computation using the linear representation of parallelism (based on the fork join and quit constructs) are as follows:

Fork L4

P1

Fork L2

Fork L3

P11

P12

J1 join(3)

P5

Fork L6

Fork L7

P8

J2 join(3)

P9

P10

quit()

L4: P4

Quit

L2: P2

goto J1

L3: P3

goto J1

L6: P6

goto J2

L7: P7

goto J2

**Answer 4:**

1. **Parallelism Profile:**

**b. Shape:**

Pi = Prob(P(t) ≤ i)

P1=8/56= 0.142

P2= (34+8)/56= 0.75

P3= (34+8+14)/56= 1.00

**c. Maximum Parallelism:** Maximum Parallelism is the maximum number of processors that may be active when an unbounded number of processor is available. The formula to calculate the value of maximum parallelism is :

m = maxi P(t), so the value of m will be 3.

**d. Fraction of Sequential Work:** The fraction of sequential work is calculated by the formula:

Fraction of Sequential Work (F) = Time spent running only 1 processor / Total time of all processor

F= 8x1 / (8x1+ 34x2 + 14x3)

F= 8/118

F= 4/59

F=0.677

**Answer 3:**

**a.** To draw the shape of the application, we need to calculate

Shape: Plot of Pi vs. i where Pi = Prob(p(t) ≤ i)

P1 = 4/12 = .333

P3= 9/12= .75

P5= 12/12= 1.0

The shape of the application would be Pi vs i graph

|  |  |
| --- | --- |
| i | P(i) |
| 1 | 0.333 |
| 3 | 0.75 |
| 5 | 1.0 |

**b)**

We know: S(N) =

and

T(N) = T(∞) []

T(∞) = 12 because the total time the given application runs is 12 sec.

To calculate the speedup , we first calculate

T(1)=12[.333+(3\*.4166)+(5\*.25)]= 33.9936

T(2)= 12 [(.333+((3/2)\*.4166)+((5/2)\*.25)]=18.994

T(3)=12 [.333+.4166+((5/3)\*.25)]=13.9952

T(4)= = 12[.333 +.4166+((5/4)\*.25)]=12.745

T(5)=12[ .333+.4166+.25]=11.9952

And

S(1)=1

S(2)=T(1)/T(2)=1.7897

S(3)=T(1)/(T(3)=2.4289

S(4)=T(1)/T(4)=2.667

S(5)= T(1)/T(5)=2.8339

|  |  |
| --- | --- |
| number of processors | Speedup |
| 1 | 1 |
| 2 | 1.7897 |
| 3 | 2.4289 |
| 4 | 2.667 |
| 5 | 2.8339 |

**c)** The average parallelism

A = p1\* 1+ p3\*3 + p5\*5 = (.333)+(3 \* 4166) + (5\* .25) = 2.8328

**d)** The upper bound will be the average parallelism A in this case, (except for when number of processor = 1. In that case the upper bound will be 1)

To calculate the lower bound , we use the following formula

L=NA/(N+A-1)

|  |  |  |  |
| --- | --- | --- | --- |
| number of processors | Speedup | U=Min(N,A) | L=NA/(N+A-1) |
| 1 | 1 | 1 | 1 |
| 2 | 1.7897 | 2 | 1.478 |
| 3 | 2.4289 | 2.8328 | 1.76 |
| 4 | 2.667 | 2.8328 | 1.942 |
| 5 | 2.8339 | 2.8328 | 2.07 |

**Answer 5:**

We know that if average parallelism A is known,

L<=S(N)<=U

where L = NA/N+A-1

and U= min(N,A)

It is given that , the estimate E is equal to the middle point between U and L.

**let us assume first U=A**

E= ((NA/N+A-1)+A)/2

= (2NA+A^2 -A)/2(N+A-1)------------------------------------------------(1)

The relative error is the ratio between |E-S(N)| and S(N)

We write ,

Relative Error = |E-S(N)|/S(N) --------------------------------------------(2)

**The exact S(N) should be a point in between U and L as**

L<=S(N)<=U

We assume for the calculation that , in extreme case S(N) is equal to U or L.

**As E is the mid point between U and L , the Numerator in equation 2 (**|E-S(N)|) **produces same result for both the cases where S(N) = U and S(N) = L , but the total result maximizes if the Denominator (**S(N)) is L , meaning S(N)= L= NA/N+A-1

So, The maximum value of relative error will be achieved if S(N)=Lputting the value of E (from equation 1) and S(N)= L= NA/N+A-1in equation 2

E= |[(2NA+A^2 -A) / 2(N+A-1) - NA / N+A-1] / [NA / N+A-1]|

=|A(A-1) /2 NA|

=|A-1/2N| --------------------------------------------------------------(4)

**Similarly , if we assume U=N** ,

E= ((NA/N+A-1)+N)/2

= (2NA+N^2 -N)/2(N+A-1)------------------------------------------------(3)

Putting the value of E (from equation 1) and S(N)= L in equation 3

E=| [(2NA+N^2 -N) / 2(N+A-1) - NA / N+A-1] / [NA / N+A-1] |

=| N-1/2A | --------------------------------------------------(5)

if we put the value of A and N in equation 4 or 5 we get the maximum value associated with E