15 NOV. 2019

Concurrency keywords:

1. SPAWN = A new thread will do this line!

2. SYNC = wait for all threads I started to return.

3. NEW (variable name) = variable only accessible to that thread

4. PARALLEL for i=. each i will be a new thread.

Analysis: T_1 = Work = non-concurrent runtime

T_∞ = Span = runtime, assuming as many threads as needed.

T_P = runtime on P processors

Thm On a computer with P processors, a greedy scheduler executes a multi-threaded computation with T_1 work and T_∞ span in time

$$T_P \le T_1/P + T_\infty$$

proof: book. □

Cor The actual runtime is within a factor of 2 of optimal.

$T_* =$ optimal w/ P processors.

Proof: $T_* \le T_P \le T_1/P + T_\infty \le T_* + T_*$

$= 2T_*$ □

In-Class Exercise 10

CSCI 432

28 October 2019

Group Number:
Group members present today:

Concurrent Programming

1. What is the difference between concurrency and parallelism? (Feel free to use the internet if you are unsure).

2. What are the possible return values of the following algorithm? What is the expected return value?

Algorithm 1 ComputeX

```
Input: $\emptyset$
Output: $x$, an integer
x = 0
for PARALEL $i = 1$ to 3 do
    $x = x + 1$
end for
return $x$
```

"the goal": 3 (probably intended)

Could return: 1, 2, 3

Ways to return 3: r1, w1, r2, w2, r3 w3

$$\underbrace{r2w2}_{3}\ \underbrace{r1w1}_{2}\ \underbrace{r3w3}_{1} = 6$$

3. Above is an example of a *race condition*, where running concurrent threads could result in multiple outputs. Explain an example application where this could be problematic.