Nicole Deyerl
Parallel computing
HW 1
2/6/17

① – What fraction of a program must be parallelizable to maintain 95% efficiency for process numbers $p = \{10, 50, 100\}$

$$\text{parallel speedup} = \frac{t}{\text{parallel exec. time}} \qquad t = \text{sequential execution time}$$

$$= \frac{t}{t(f + \frac{1-f}{p})} \qquad t = \text{time on one processor}$$

$$= \frac{1}{f + \frac{1-f}{p}}$$

$$\text{efficiency} = \frac{\text{parallel speedup}}{\text{processors used}}$$

$$= \frac{1}{f + \frac{1-f}{p}} \cdot \frac{1}{p}$$

$$= \frac{1}{pf + 1 - f}$$

$$.95 = \frac{1}{pf + 1 - f} \Rightarrow f = \frac{1}{19(p-1)}$$

$$p = 10: \quad f = \frac{1}{19(9)} = \frac{1}{171} = .0058 \qquad .58\%$$

$$p = 50: \quad f = \frac{1}{19(49)} = \frac{1}{931} = .00107 \qquad .107\%$$

$$p = 100: \quad f = \frac{1}{19(99)} = \frac{1}{1881} = .00053 \rightarrow .053\%$$

(2) - $\frac{1}{k}$ of processing algorithm must be done in serial

- max possible parallel speedup

$\Rightarrow f = \frac{1}{k}$

as above speed up $= \dfrac{1}{f + \frac{1-f}{p}}$

$$= \frac{1}{\frac{1}{k} + \frac{1-1/k}{p}}$$

$$= \frac{1}{\frac{k+p-1}{kp}} \quad \text{by wolfram}$$

$$= \frac{kp}{k+p-1} \quad \text{max speed up}$$

(3) - task must be broken into m subtasks
  - each requires one unit of time to perform
- how much time needed for m-stage pipeline to process n subtasks
  - no overhead due to communication

we can execute m-stages at once using an m-stage pipeline, so processing n subtasks will take $\sim \frac{n}{m}$ time units

④ - current computer: solves problem w/ size $n = 10^{16}$ in 16 hours
- CPU speed only

- how large a problem can be solved in 16 hours by a computer that is 100 times faster if the programs complexity is

a) $O(n)$ b) $O(n \log n)$ c) $O(n^2)$ d) $O(n^3)$

$$\text{time} = \frac{\text{work}}{\text{rate}} = \frac{\text{complexity}}{\text{rate}}$$

initial (current computer): $16 = \frac{10^{16}}{r} \Rightarrow r = 6.25 \times 10^{14}$

a) $16 = \frac{n}{100 \cdot r} \Rightarrow n = 16 \cdot 100 \cdot r$
$= 1 \times 10^{18}$

b) $16 = \frac{n \log n}{100 \cdot r} \Rightarrow n \log n = 16 \cdot 100 \cdot r = 1 \times 10^{18}$
$\Rightarrow n = 2.6 \times 10^{16}$ via wolfram alpha, solving $n \log = 1 \times 10^{18}$ for n

c) $16 = \frac{n^2}{100 \cdot r} \Rightarrow n^2 = 1 \times 10^{18}$
$\Rightarrow n = 1 \times 10^{9}$

d) $16 = \frac{n^3}{100 \cdot r} \Rightarrow n^3 = 1 \times 10^{18}$
$\Rightarrow n = 1 \times 10^{6}$

⑤ - superlinear speedup: parallel program w/ speedup > p

- 2 other examples of how a program might overcome a resource limitation to achieve superlinear speedup.

1) A program could overcome a resource limitation if we split a serial code p ways to be performed on p processors, while also improving the way the code is written by doing something like changing how the data is stored or just improving the code to make it more efficient.

2) A serial code could be spending a lot of time moving data back and forth between the CPU, disk and RAM because of memory getting used up if you're just using 1 processor. In parallel, you get the benefit of p processors working simultaneously, and there is less data per processor so memory won't become full and only communication between the CPU + cache has to take place.

⑥ - 18 students are throwing a birthday party for the professor

a) - tasks that can be assigned allowing students to use task parallelism

Each student works on one of the following tasks:

- baking the cupcakes
- decorating the cupcakes
- making decorations
- hanging decorations
- making the birthday card
- signing the birthday card
- making invitations
- sending out invitations

+10 more tasks for 18 students total

- schedule showing when various tasks can be performed (in case data dependencies exist)

① Bake cake ② Decorate cake	① make birthday card ② sign birthday card → pipeline
① Make decorations ② Hang decorations	① make invitations ② send out invitations

b) How to use data-parallelism to clean the house before the party begins

Each student cleans 1/18th of the room from floor to ceiling.

c) - combine both approaches to use a combo of task + data parallelism so each student does the same amount of work

Have assembly line stations in the room for groups of students to work on chunks of a task. For example, at the baking station, one student will mix batter, one will cook the cakes, one will mix frosting, one will decorate. At the decorating station, one student will cut out decorations, another will color, and another will hang them.

(7) - need to compute sum of 1000 numbers as quickly as possible
- each # is on an index card, I hold the stack
- in charge of 200 accountants, each w/ pencil + own stack of index cards
- accountants sit in rows of 10 x 20
- accountants can:
  - add 2 #s at a time + write the result on a blank card → 1s
  - pass any # of cards to 4 nearest accountants → 3s

a) - Fast method to distribute cards to accountants

A fast method to distribute cards evenly to accountants would be to give 250 cards to the 4 accountants in each corner, then have every accountant keep 5 cards and evenly distribute whatever amount of cards are left to the 4 accountants nearest to them. Then each of those accountants should keep 5 cards and evenly distribute their remaining cards to the (max) 4 accountants nearest to them who don't have cards.

b) - Fast method to accumulate subtotals generated into a grand total

The accountants should each total the sum of the 5 cards in their posession. Then, they should pass their (now 1) card back toward the 4 corners, and the 4 corner accountants will subtotal the cards as they recieve them. Then I will gather the 4 corner subtotals and add them into a grand total.

c) - time needed to perform

a) + 4s to pass out 4 stacks in 4 corners

+ ~50 passes to 4 nearest neighbors until all cards are distributed (done simultaneously by n accountants starting from 4 corners)

$\Rightarrow 50 \times 3s \times \frac{200}{n} = \left(\frac{30,000}{n}\right)$ seconds

$\Rightarrow \frac{30,000}{n} + 4$ s

$n = 200 \Rightarrow 154$ s $\qquad n = 10 \Rightarrow 3,000$ s

$n = 50 \Rightarrow 604$ s.

b) $+ 3s \cdot \frac{200}{n}$ to add 5 cards (done simultaneously by n accountants) (5 cards → 3 sums → 3 s)

10×5 → max # of desks an accountant could be from a corner accountant

$+ 50 \cdot 3s \cdot \frac{200}{n}$ max time for ∀ cards to get passed to corner accountants (done simultaneously by n accountants)

+ 0s for corner accountants to subtotal, because this is happening simultaneous to the passing and there will be no back-up since adding is faster than passing.

+4s for me to collect the 4 subtotals

+2s for me to add them up

$\Rightarrow \frac{600}{n} + \frac{30{,}000}{n} + 6$ s

$n = 200 \Rightarrow 159$ s $\quad n = 10 \Rightarrow 3066$ s

$n = 50 \Rightarrow 618$ s

d) For the same 3 numbers of active accountants, estimate total time to do 10,000 numbers w/ same 200 accountants

a) Each accountant should take 50 cards instead of 5.

$\Rightarrow$ doesn't affect time because accountants can pass an unlimited number of cards at once.

$n = 200 \Rightarrow 154$ s $\quad n = 10 \Rightarrow 3{,}000$ s

$n = 50 \Rightarrow 604$ s.

b) Each accountant has to add 50 cards total instead of 5

$\Rightarrow$ addition time: $3s \cdot \frac{200}{n} \Rightarrow 25s \cdot \frac{200}{n} = \frac{5000}{n}$ s

since 2 cards can be added at once

Everything else stays the same since they can pass an unlimited # of cards at once, corner accountants add faster than the cards are getting passed, and in the end I still only have 4 cards to total.

$$\Rightarrow \frac{5000}{n} + \frac{30{,}000}{n} + 6 \text{ s}$$

$n = 200 \Rightarrow 181$ s $\quad n = 10 \Rightarrow 3506$ s

$n = 50 \Rightarrow 706$ s

e) 1000 accountants cannot complete a task 1000 times faster than 1 accountant because they need to share data with each other, which takes extra time. If the accountants could all instantaneously download data then instantaneously upload it after working on it, the task would be completed 1000 times faster at best (assuming each of 1000 accountants work @ same rate as orig. accountant). Instead, time is eaten up passing information.

f) A better arrangement would be a circle because the max distance an accountant could be away from a submanager (the distributor / subtotaler in the corner) is reduced. This way, passing in/out the cards, which was responsible for a majority of the time, is lessened.