Comp 212: Functional Programming

Fall 2022

Dan Licata

Lecture 1: Parallelism

Count how many

People took

COMP 211

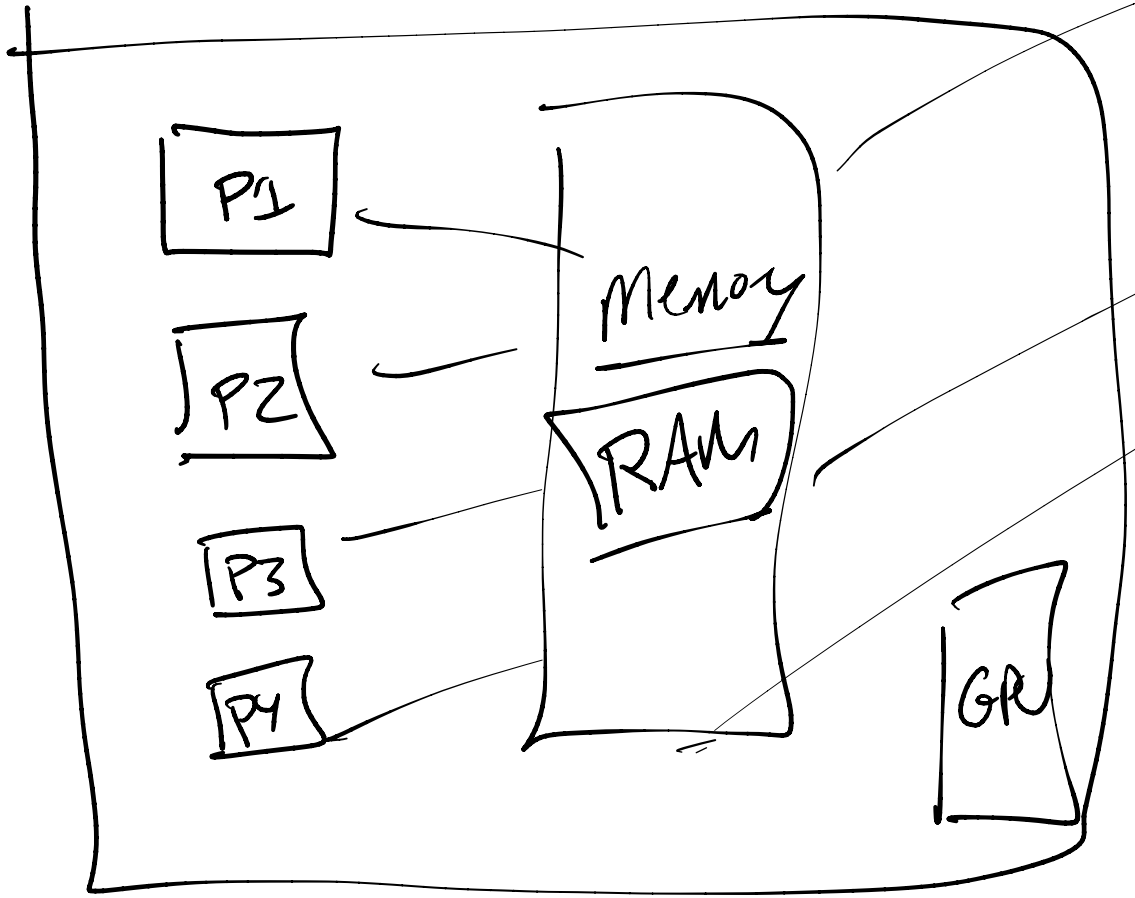
in C

No parallelism ⟶ "sequentially"

Parallel ⟶ many computations
at
the same time

Hardware

"multicore"



Hard disk

Processor:

adds
mults
subtract
bitwise
memory

Moore's law: processors got faster (2x every 18 months)

Multicore era: can do more things at once

2
4
16
32
64

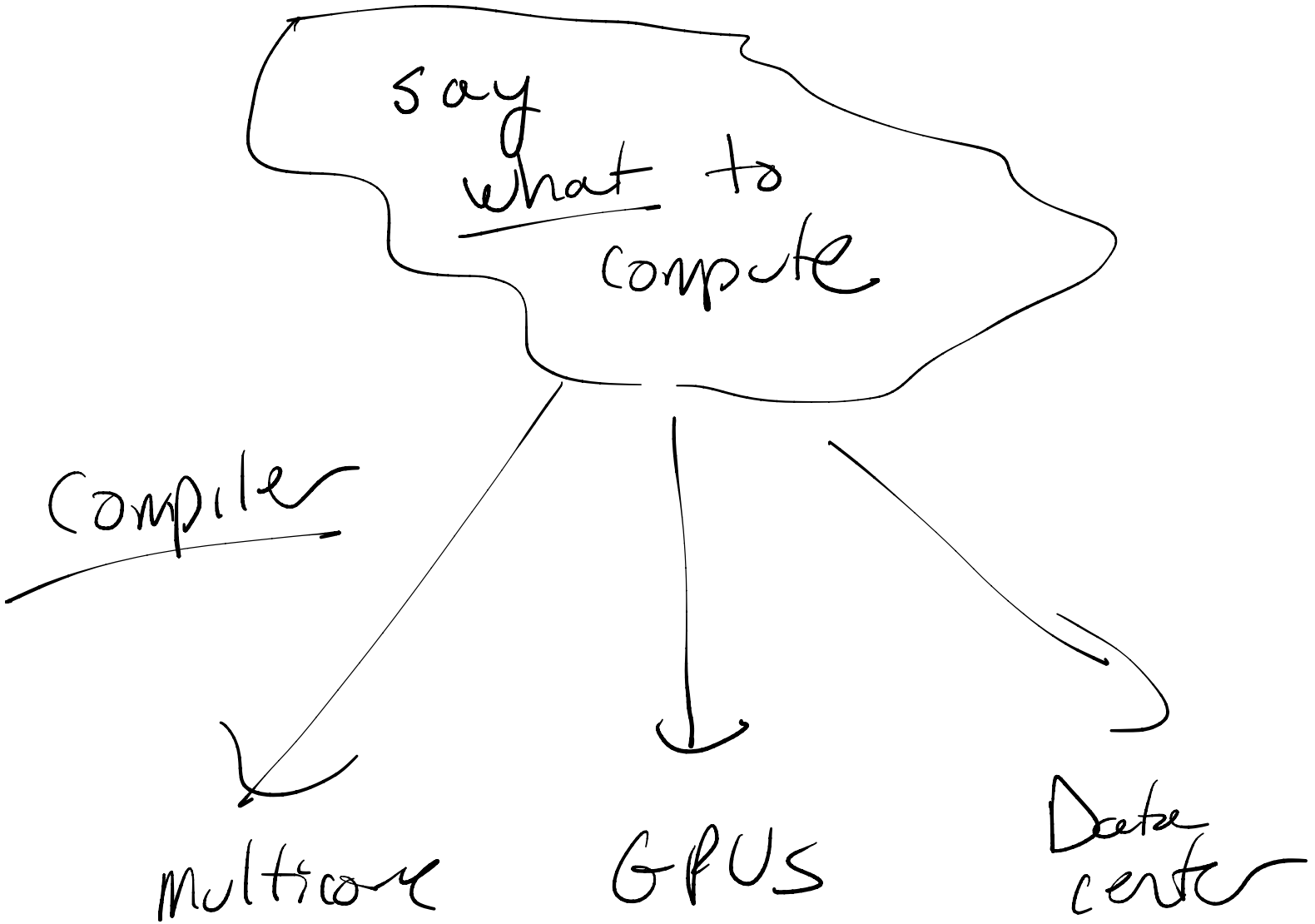
"cores" / processor

GPU / graphics processing unit

→ 1000s - 4000s

Data center:

10,000s of computers networked together


say
what to
compute
Compiler
multicore
GPUs
Data
center

Course goals: you will learn to

(1) write parallel functional programs

(2) analyze their parallel and sequential running time

(3) reason mathematically about the correctness of them

(4) (abstract types)

(* Purpose: count # of people
who took COMP 211 in (*)

(* Example: count(<<yes, no, yes>, <no, yes, no>>) = 3 *)

count(<<1, 0, 1>, <0, 1, 0>>)

sequence of int sequences

1 = yes
0 = no

Algorithm:

1. Sum each row (in parallel)
2. Sum the column that we get at the end

type row = int sequence ⇝ e.g. ⟨1,0,1⟩

type class = row sequence
= (int sequence) sequence } ⟨⟨1,0,1⟩, ⟨0,1,0⟩⟩

fun sum(r: row) : int = ...
(e.g. sum ⟨1,0,1⟩ = 2)

fun count(c: class) : int =

sum (map(sum, c))

step 2 step 1

count ⟨⟨1, 0, 1⟩, ⟨0, 1, 0⟩⟩

computing by calculation

⟶ "step" sum ⟨ sum⟨1, 0, 1⟩, sum⟨0, 1, 0⟩⟩

⟶ in parallel sum ⟨ 2, 1 ⟩

⟶ 3

Predict (parallel) running time on big inputs

Work | sequential running time

$n \times n$ classroom

How long to count?

$O(n^2)$ $\approx n^2$

(proportional to)

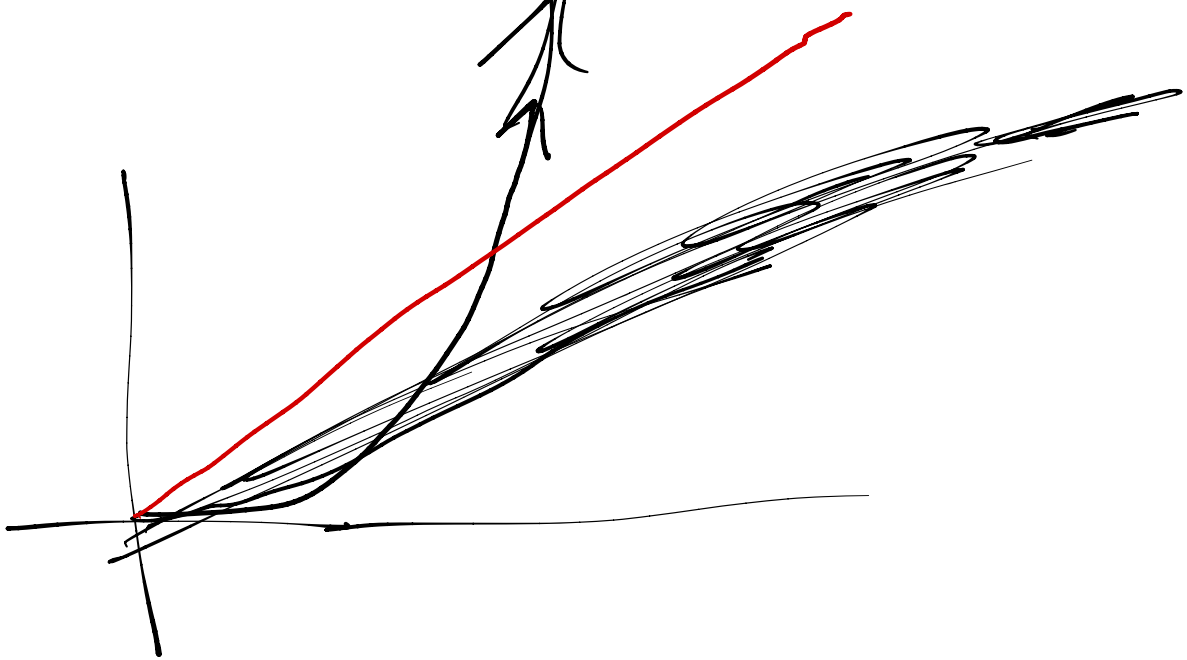
span / parallel running time
on "enough" processors

$n \times n$

① add up each row
in parallel $O(n)$

② add final column $O(n)$

$O(2n)$
$= O(n)$



Work and span predict running time on fixed P processors

time on P procs is proportional to $\max\left(\frac{work}{P}, S\right)$

Work = 100
Span = 10
} 10 × 10

P = 2

$$\max\left(\frac{\text{work}}{P}, \text{span}\right)$$

P1 P2

$$= \max\left(\frac{100}{2}, 10\right)$$

$$= \max(50, 10) = 50$$

P = 100

$$\max\left(\frac{\text{work}}{P}, \text{span}\right)$$

$$= \max\left(\frac{100}{100}, 10\right)$$

$$= \max(1, 10) = 10$$

Prove:

for all int sequences r

sum(r)

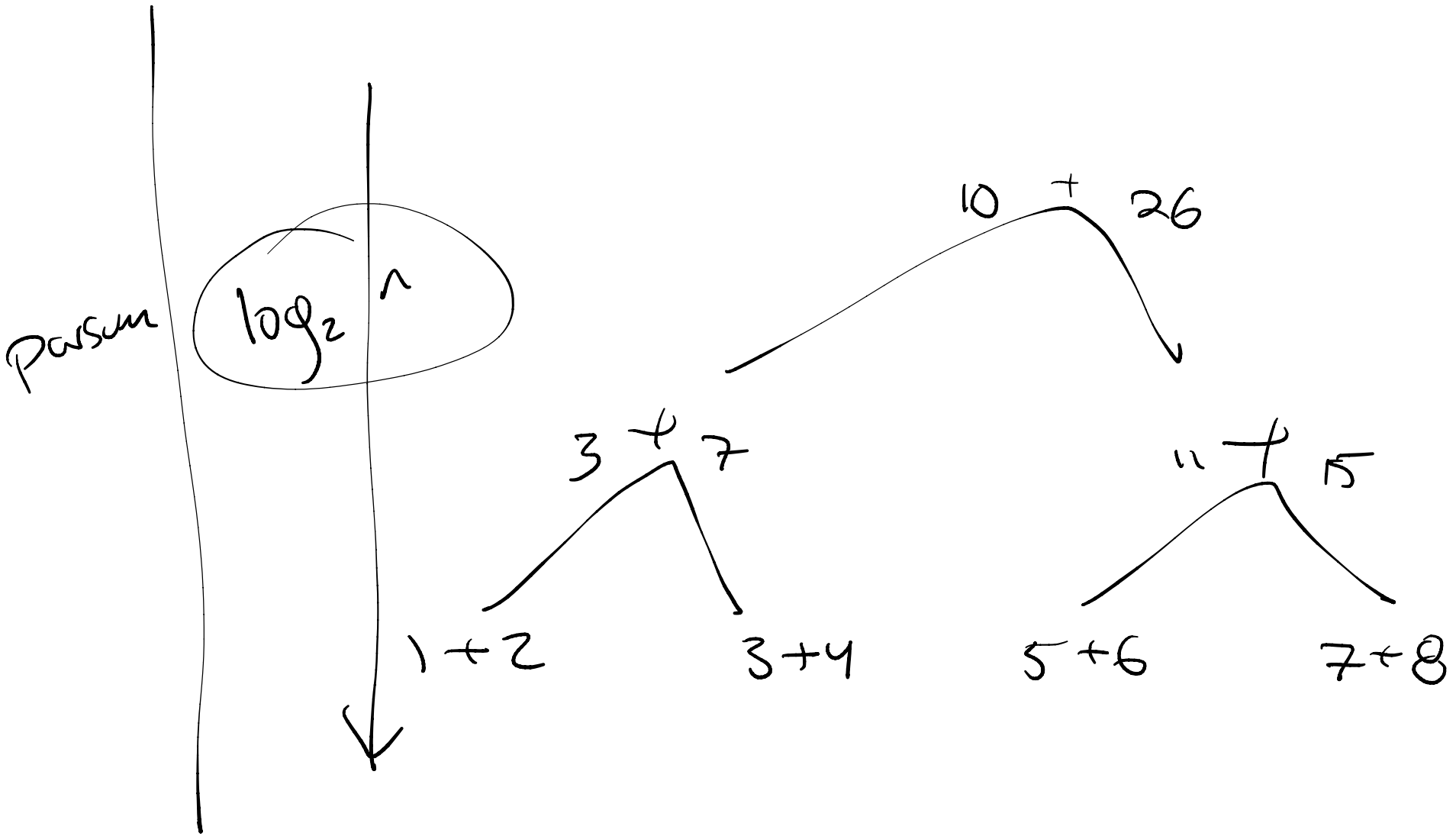
= (computes the same number as)

parsum(r)

↳ parallel summing

$1 + (2 + (3 + (4 + (5 + (6 + (7 + 8))))))$

Sum



Activities

1. Lecture
2. Labs
3. Homework → regular 80%
   → challenge 20%
4. TA sessions at night