Lecture 4: Intro to Parallel Computing

CMSE 822: Parallel Computing

Prof. Sean M. Couch

Why compute in parallel?

- Need more memory (big problem)
- Need to go faster
- Both

Question:

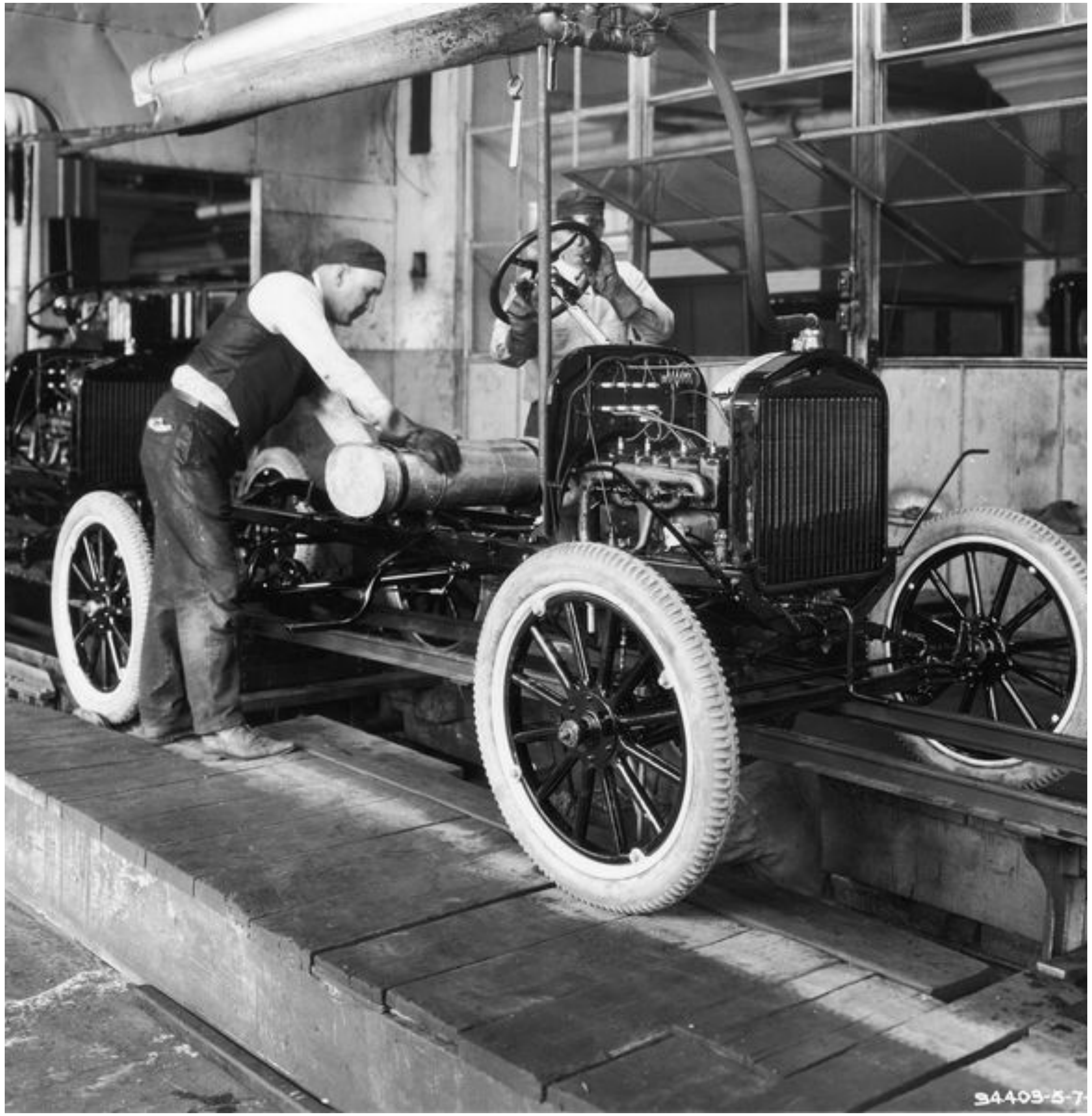
If I run a problem on *p* processors will it run *p* times faster than on one processor?

- Yes
- No
- It depends…

Kinds of parallelism

Serial

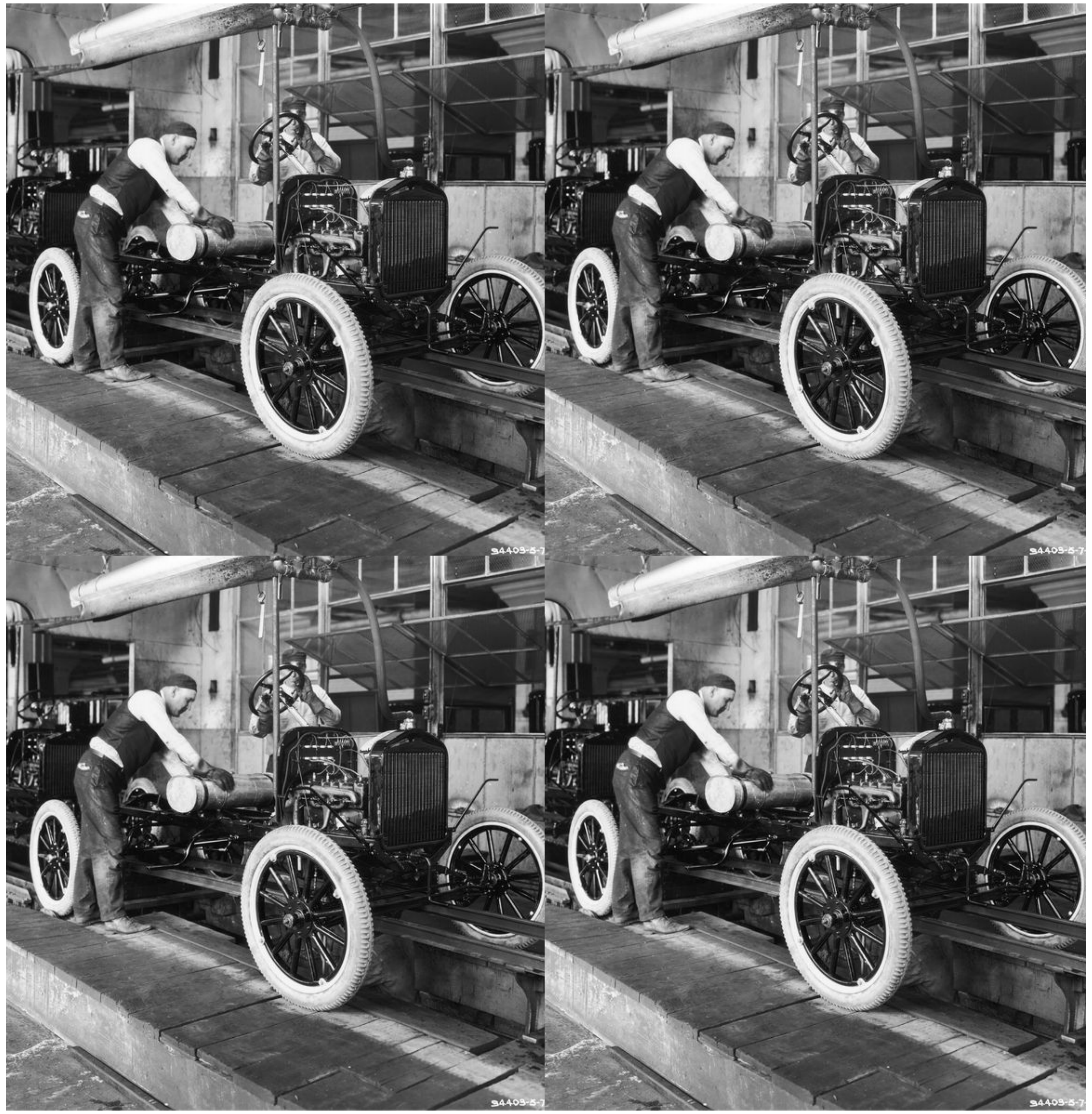
- One worker builds one car, one at a time, do all steps necessary to assemble the car
- In Flynn’s taxonomy, what of computation is this?
  - SISD
  - SIMD
  - MISD
  - MIMD



Kinds of parallelism

Data parallelism

- N workers build N cars simultaneously, each worker completely all steps necessary to assemble one car
- In Flynn's taxonomy, what of computation is this?
    - SISD
    - SIMD
    - MISD
    - MIMD



Kinds of parallelism

Functional parallelism

- N workers build one car, each worker completely some unique subset of the steps necessary to assemble one car
- In Flynn's taxonomy, what of computation is this?
  - SISD
  - SIMD
  - MISD
  - MIMD

Kinds of parallelism

Data/Functional parallelism

- N workers build N cars simultaneously, each worker completely some unique subset of the steps necessary to assemble one car

- Notice the analogy with communication and bandwidth represented by the moving assembly line!

Speedup, efficiency, and Amdahl

T_1 : *the time the computation takes on a single processor*
T_p : *the time the computation takes with p processors*
T_∞ : *the time the computation takes if unlimited processors are available*
P_∞ : *the value of p for which $T_p = T_\infty$*

- Speedup: $S_p = T_1/T_p.$
- Efficiency: $E_p = S_p/p.$
- Amdahl's Law:

$$T_P = T_1(F_s + F_p/P).$$

- asymptotically, $S_P \leq 1/F_s$
- with communication:

$$T_p = T_1(F_s + F_p/P) + T_c,$$

Exercise 2.10

Exercise 2.10. Let's do a specific example. Assume that a code has a setup that takes 1 second and a parallelizable section that takes 1000 seconds on one processor. What are the speedup and efficiency if the code is executed with 100 processors? What are they for 500 processors? Express your answer to at most two significant digits.

- Sequential time: $T_1 = 1001$
- With 100 processors: $T_{100} = 11$, $S_{100} = 1001/11 \sim 91$, $E_{100} \sim 0.91$
- With 500 processors: $T_{500} = 3$, $S_{500} = 333$, $E_{500} \sim 0.67$

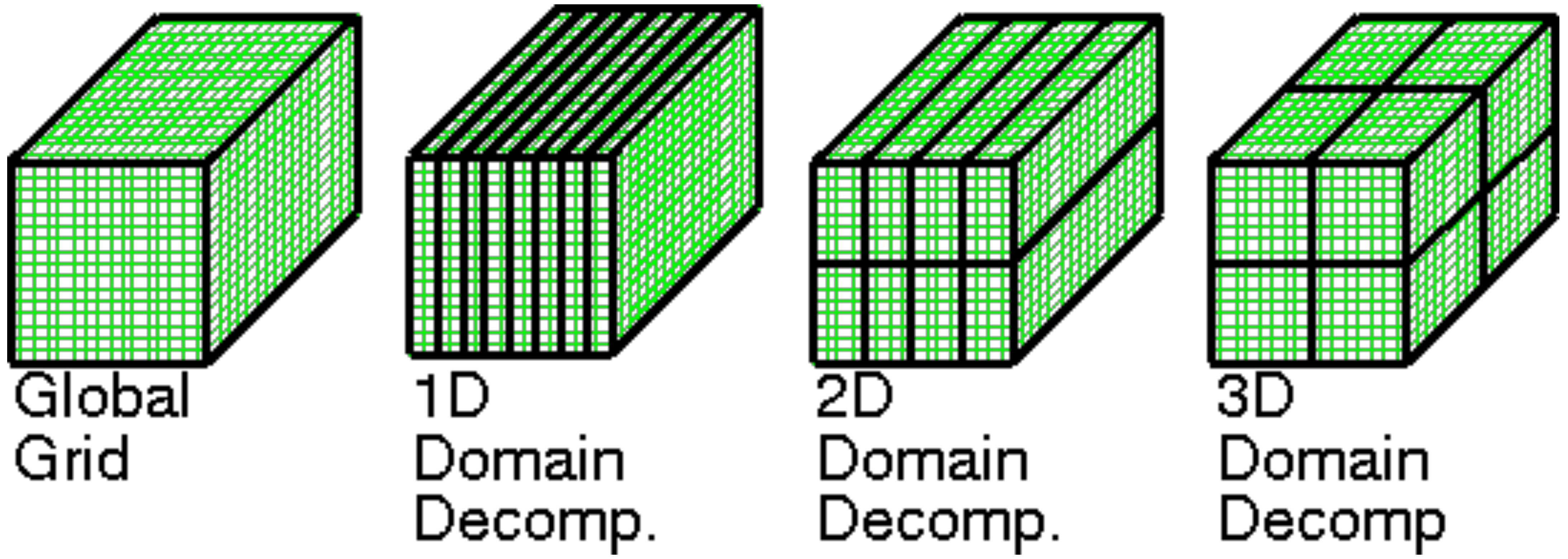
Scalability

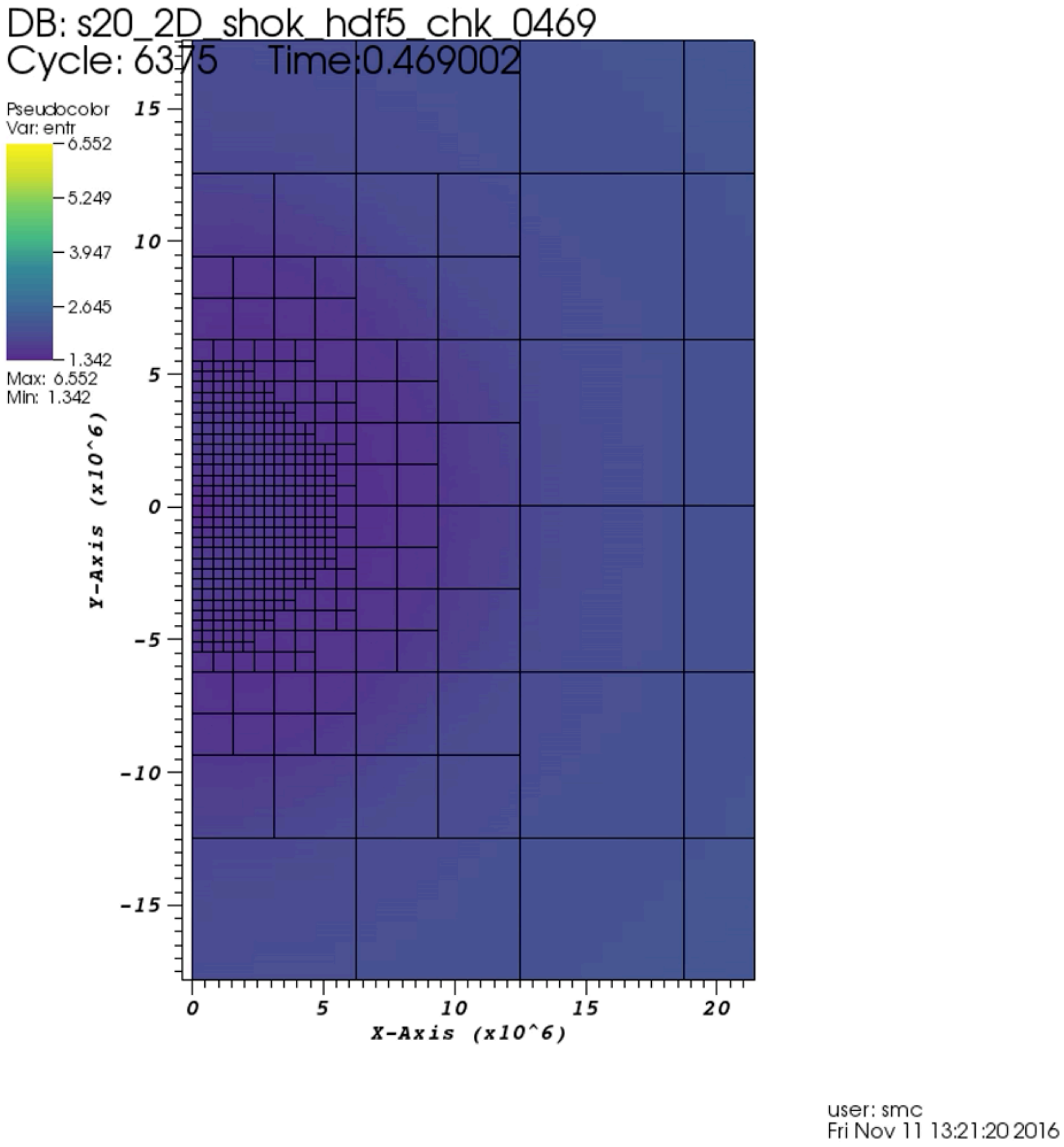
- Strong scaling: Fixed problem size, increased number of processes
- Weak scaling: Fixed problem size *per process*

Scalability

“Real” world example

- Domain decomposition in a fluid calculation

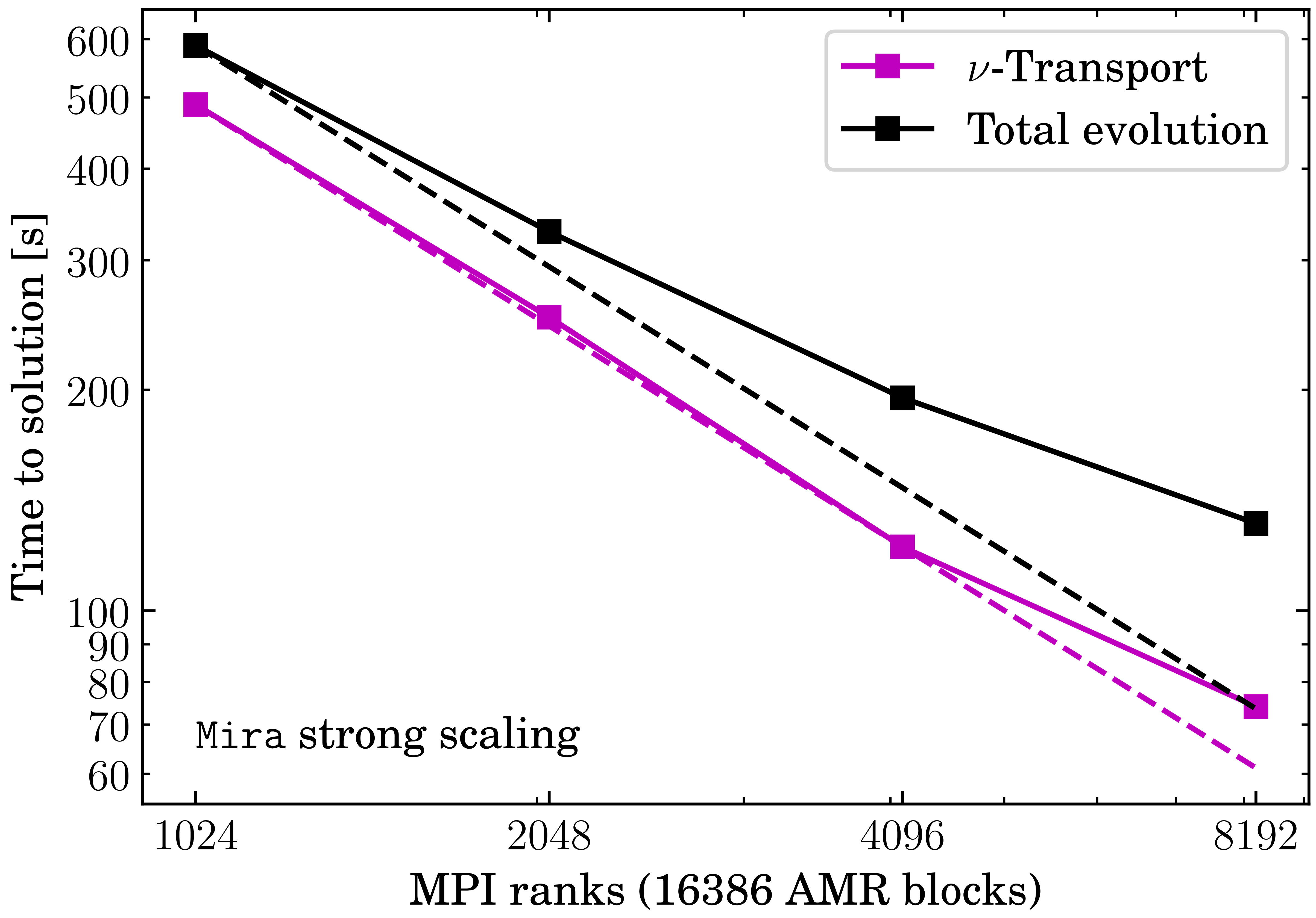




Scalability

“Real” world example

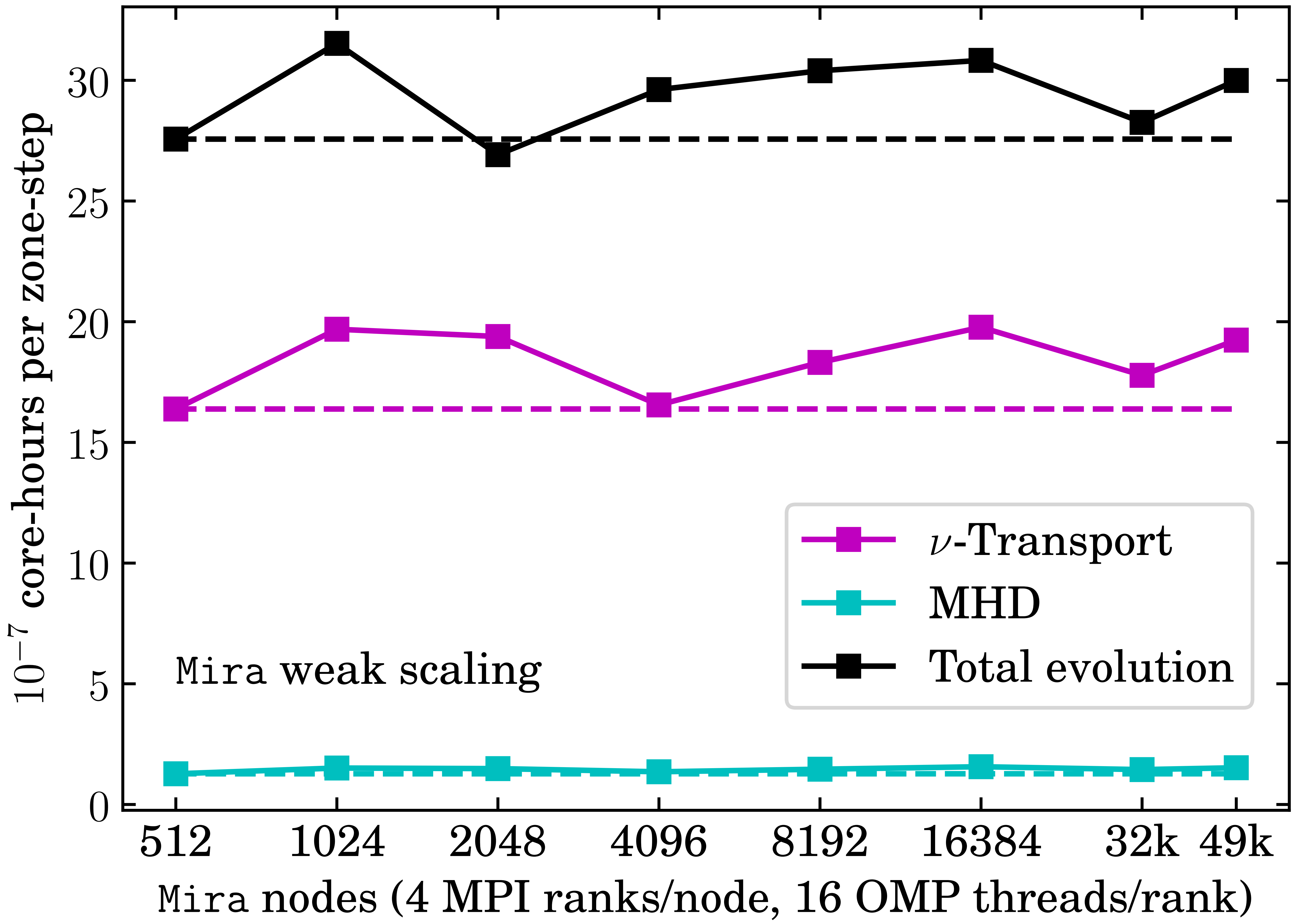
- Strong scaling: challenging!
- Hard to fit big problem on few processes
- Communication cost AND serial fraction are major limiters



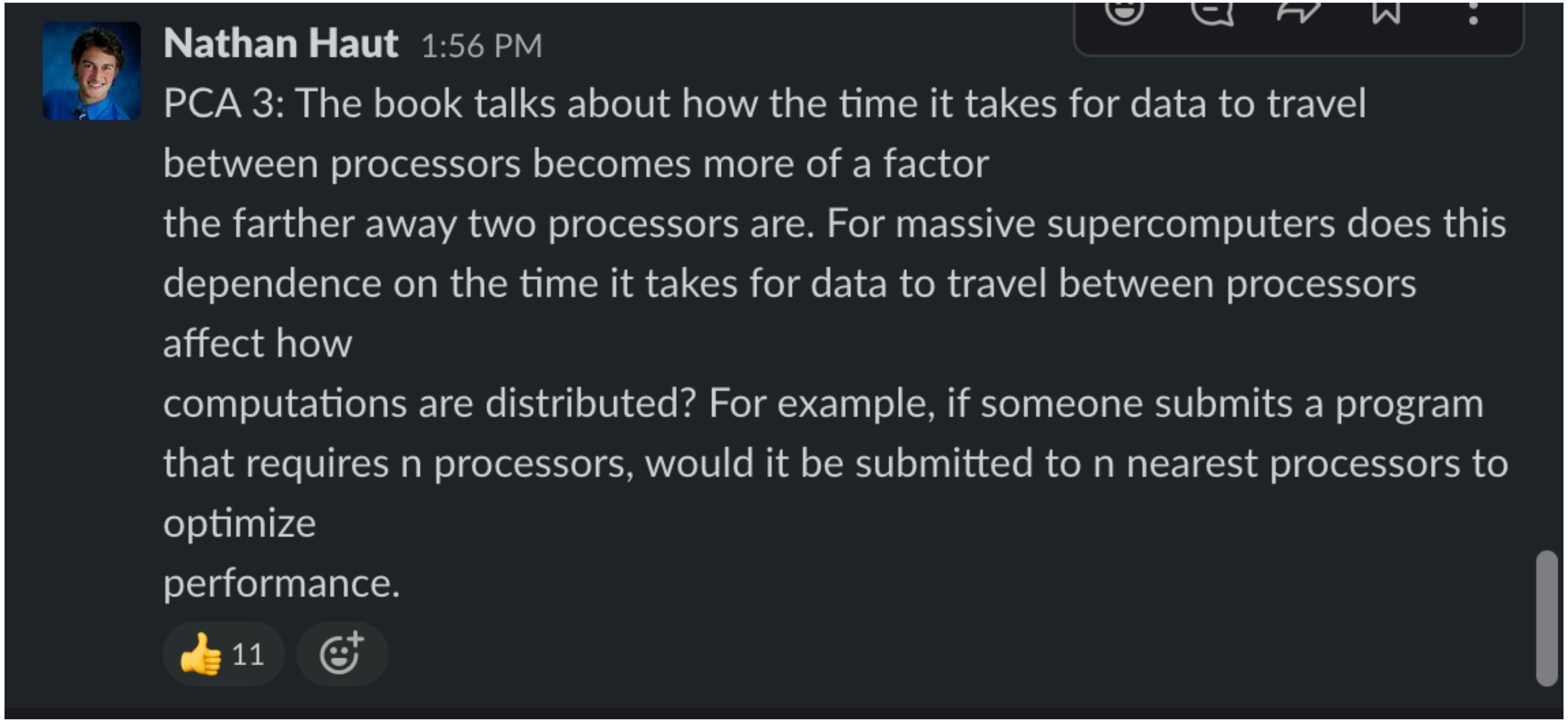
Scalability

“Real” world example

- Weak scaling: better idea of how application will perform “at scale”
- Strong and weak scaling together are needed to get a sense of real efficiency



Questions



Nathan Haut 1:56 PM

PCA 3: The book talks about how the time it takes for data to travel between processors becomes more of a factor
the farther away two processors are. For massive supercomputers does this dependence on the time it takes for data to travel between processors affect how
computations are distributed? For example, if someone submits a program that requires n processors, would it be submitted to n nearest processors to optimize
performance.

11

Questions

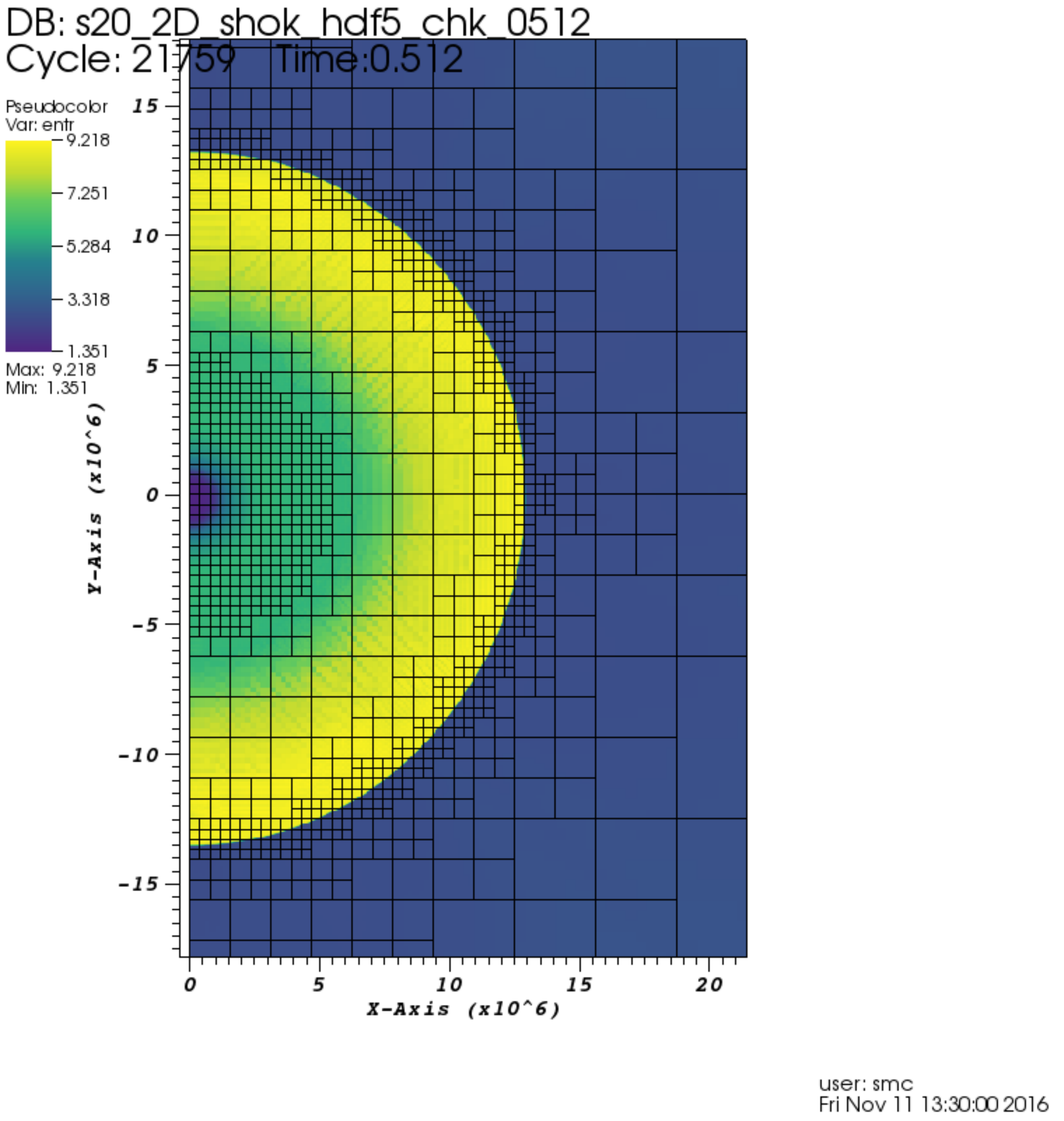
Stephen White 10:38 PM

PCA3: The book mentions deadlocks only to say it won't be discussing it much. It's also mentioned different cores messing with shared data and potentially having to update other processors, but hasn't gone into a ton of detail. Will we cover how to make sure different cores share data safely?

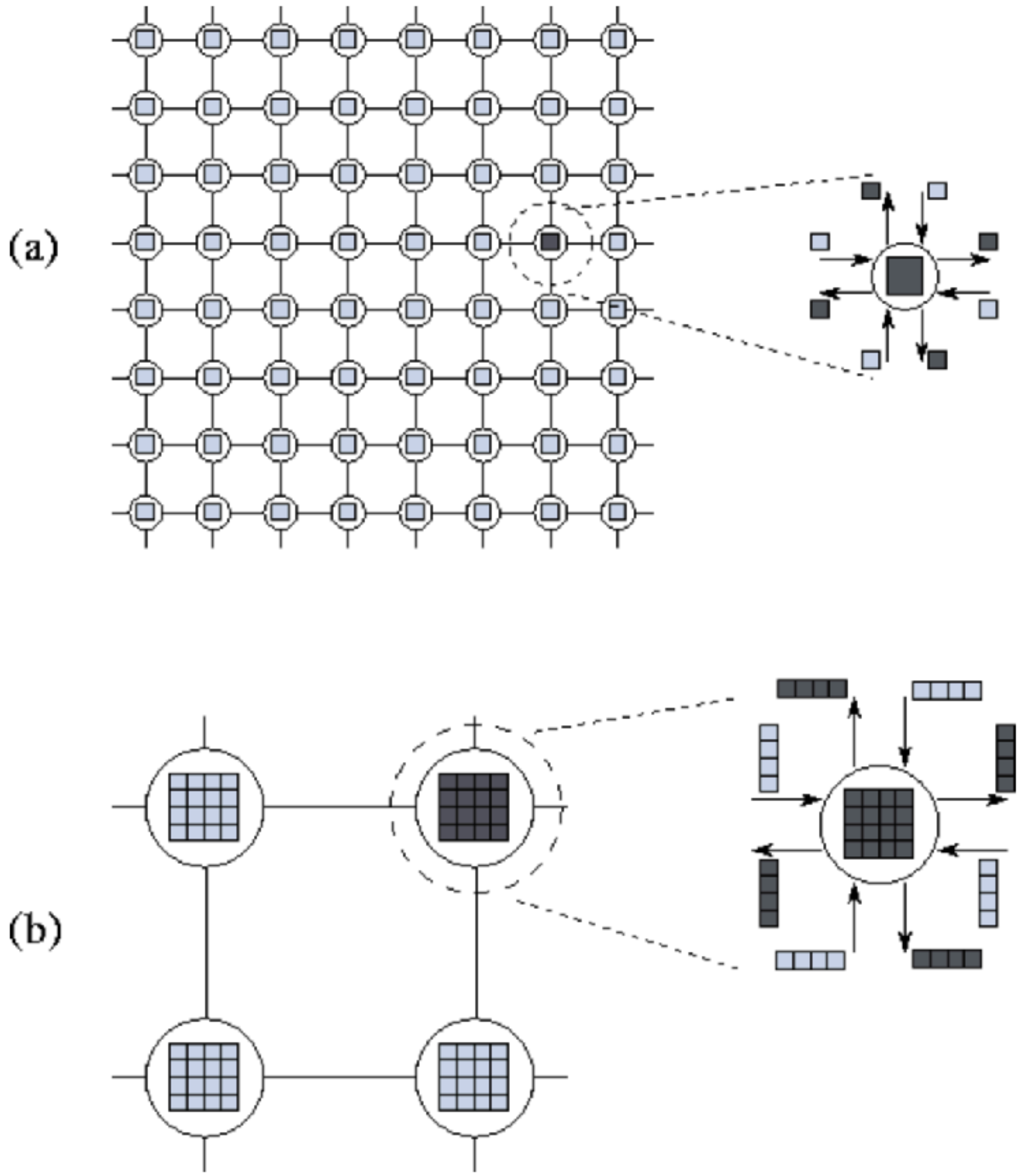
👍 6

Questions
Granularity

- “the amount of work (or the task size) that a processing element can perform before having to communicate or synchronize with other processing elements”



Questions
Surface to volume



Group work on HW2