Pollack’s rule for system modeling

2/23/26

Final exam

- Scheduled for Sunday (3/15) 8:30-11:30am

System modelling: What kinds of multicore systems should we build?

Hardware designer

- Processor capabilities determined by what circuits are provided in hardware
- For a given amount of silicon area, how should it be allocated to maximize performance?

Pollack’s rule

- The performance of a processing core is proportional to the square root of its area

Pollack's rule

- The performance of a processing core is proportional to the square root of its area

If a single core is replaced by 4 cores, each ¼ as large, what is the expected peak performance of the entire system? (i.e. the performance assuming all 4 could be kept perfectly busy)

A. Half as much as before
B. The same as before
C. Twice as much as before
D. Four times as much as before
E. None of the above

Pollack's rule

- The performance of a processing core is proportional to the square root of its area

If a single core is replaced by 4 cores, each ¼ as large, what is the expected peak performance of the entire system? (i.e. the performance assuming all 4 could be kept perfectly busy)

A. Half as much as before
B. The same as before
C. Twice as much as before
D. Four times as much as before
E. None of the above

If a single core is replaced with 4 cores, what is the running time if half the program can be parallelized?

- Parallel part:

  ½ the work / 2 the performance = ¼

If a single core is replaced with 4 cores, what is the running time if half the program can be parallelized?

- Parallel part:

  ½ the work / 2 the performance = ¼

- Serial part:

  ½ the work / ½ the performance = 1

Total time: 1.25 times as long

Recall: Amdahl's Law

(not what we're using, but similar)

$$T_p = \frac{T_1(1\text{-}B)}{p} + T_1 B$$

Time for parallel part

Time for serial part

T_p = processing time on p processors

T_1 = processing time on 1 processor

B = fraction of program that must run serially

If a single core is replaced with 4 cores, what is the running time if 75% of the program can be parallelized?

A. 0.6 times as long
B. 0.875 times as long
C. 1 times as long (the same time)
D. 1.35 times as long
E. None of the above

If a single core is replaced with 4 cores, what is the running time if 75% of the program can be parallelized?

A. 0.6 times as long
B. 0.875 times as long (0.75/2 + 0.25/0.5)
C. 1 times as long (the same time)
D. 1.35 times as long
E. None of the above

If a single core is replaced with 4 cores, what is the running time if 90% of the program can be parallelized?

A. 0.45 times as long
B. 0.65 times as long
C. 0.765 times as long
D. 1 times as long (the same time)
E. None of the above

If a single core is replaced with 4 cores, what is the running time if 90% of the program can be parallelized?

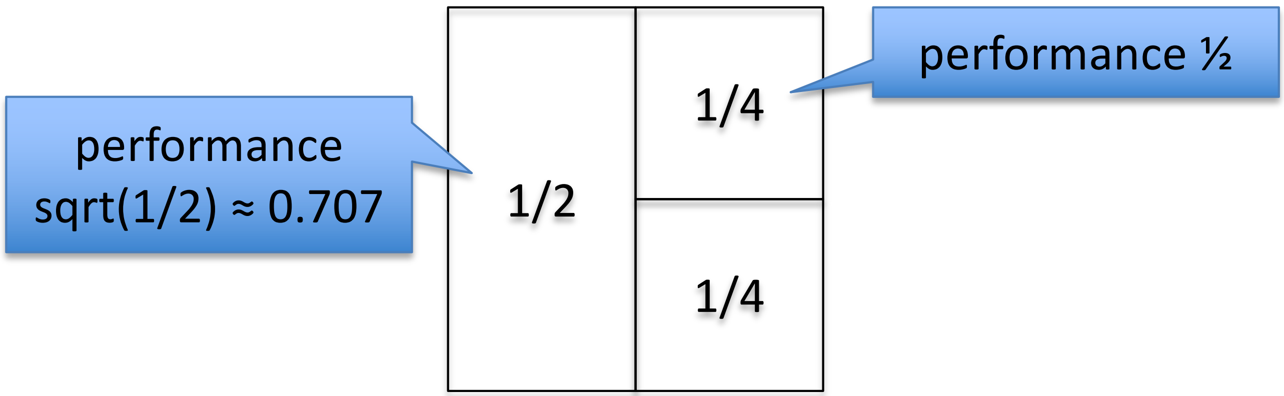
A. 0.45 times as long
B. 0.65 times as long (0.9/2 + 0.1/0.5)
C. 0.765 times as long
D. 1 times as long (the same time)
E. None of the above

Factor by which running time changes for different programs

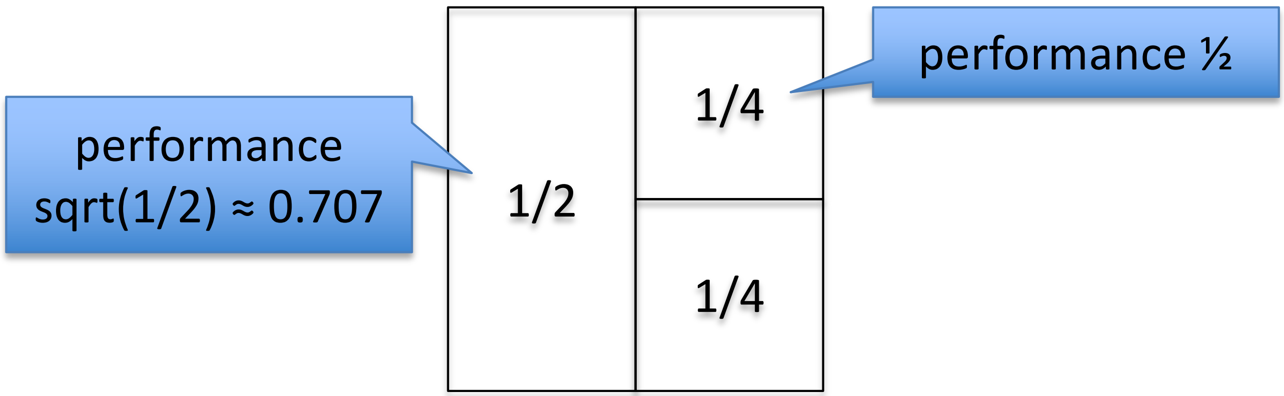
% of program that is parallelizable	75%	90%	95%
1 core	1	1	1
4 cores	0.875	0.65	0.575
9 cores	1	0.6	0.467
16 cores	1.1875	0.625	0.438
25 cores	1.4	0.68	0.44
36 cores	1.625	0.75	0.458

As the number of cores increases, highly parallelizable programs have improved performance, but less parallelizable programs suffer

What about unequal core sizes?



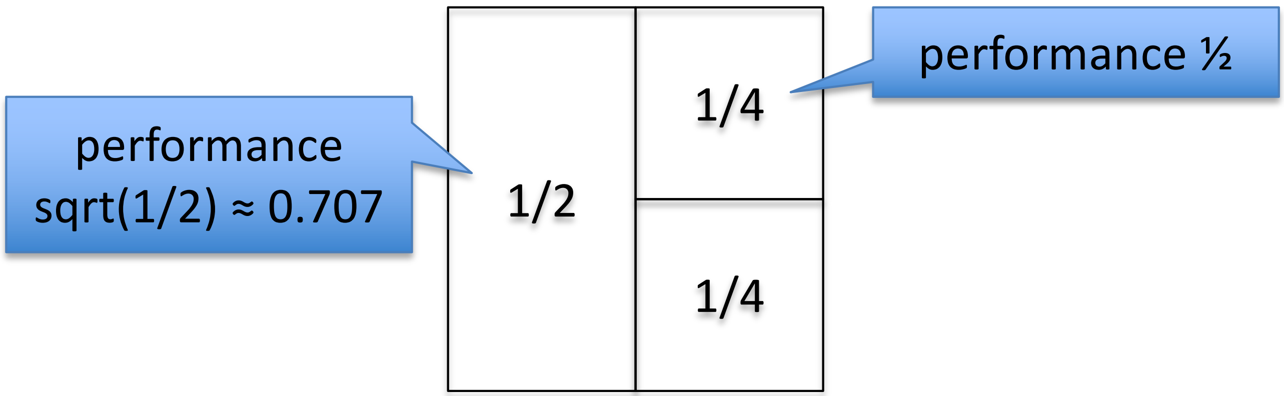
What about unequal core sizes?



If a single core is replaced with the configuration above, what is the expected peak performance of the entire system?

A. ≈0.707 times as much as before
B. ≈1.207 times as much as before
C. ≈1.707 times as much as before
D. ≈2.121 times as much as before
E. None of the above

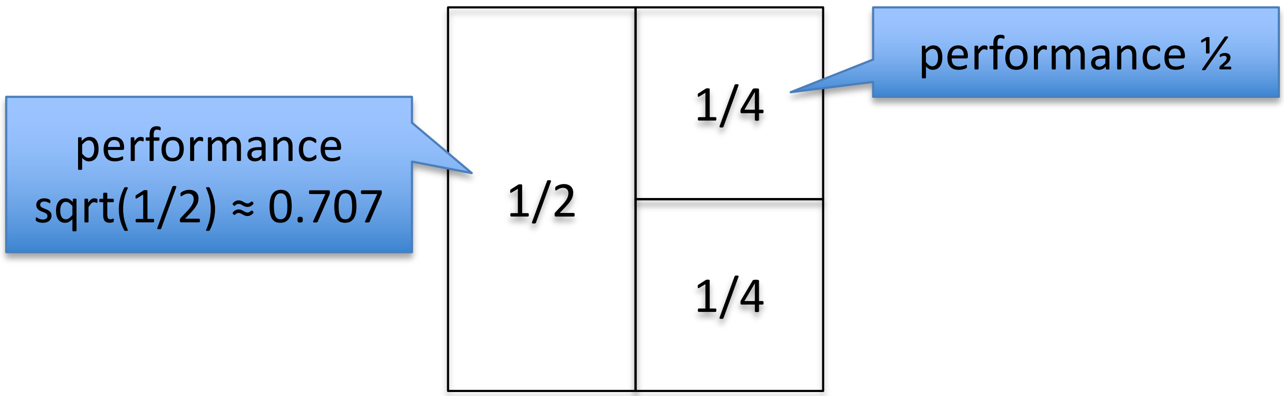
What about unequal core sizes?



If a single core is replaced with the configuration above, what is the expected peak performance of the entire system?

A. ≈0.707 times as much as before
B. ≈1.207 times as much as before
C. ≈1.707 times as much as before
D. ≈2.121 times as much as before
E. None of the above

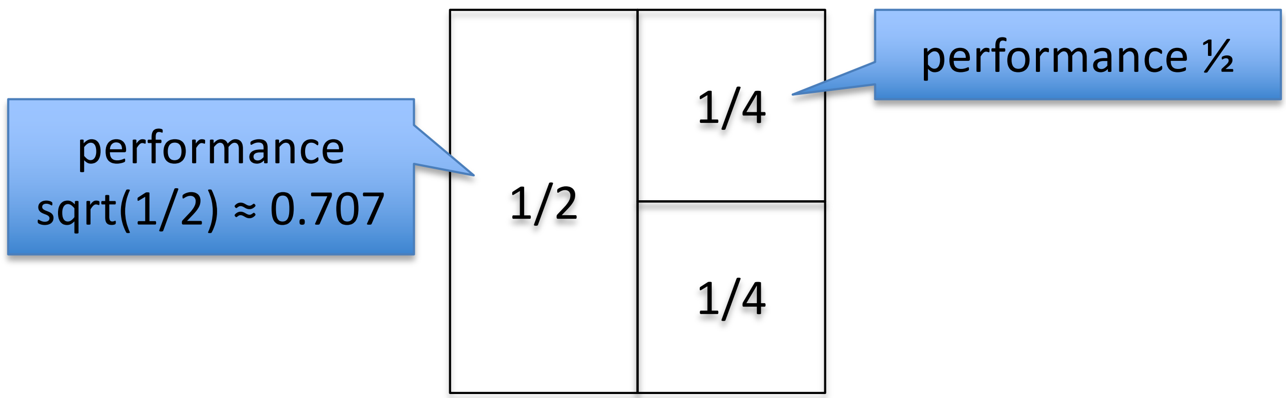
What about unequal core sizes?



What is the running time of a program that cannot be parallelized if it runs on the largest core?

A. ≈0.707 times as long
B. ≈1 times as long
C. ≈1.207 times as long
D. ≈1.414 times as long
E. None of the above

What about unequal core sizes?



What is the running time of a program that cannot be parallelized if it runs on the largest core?

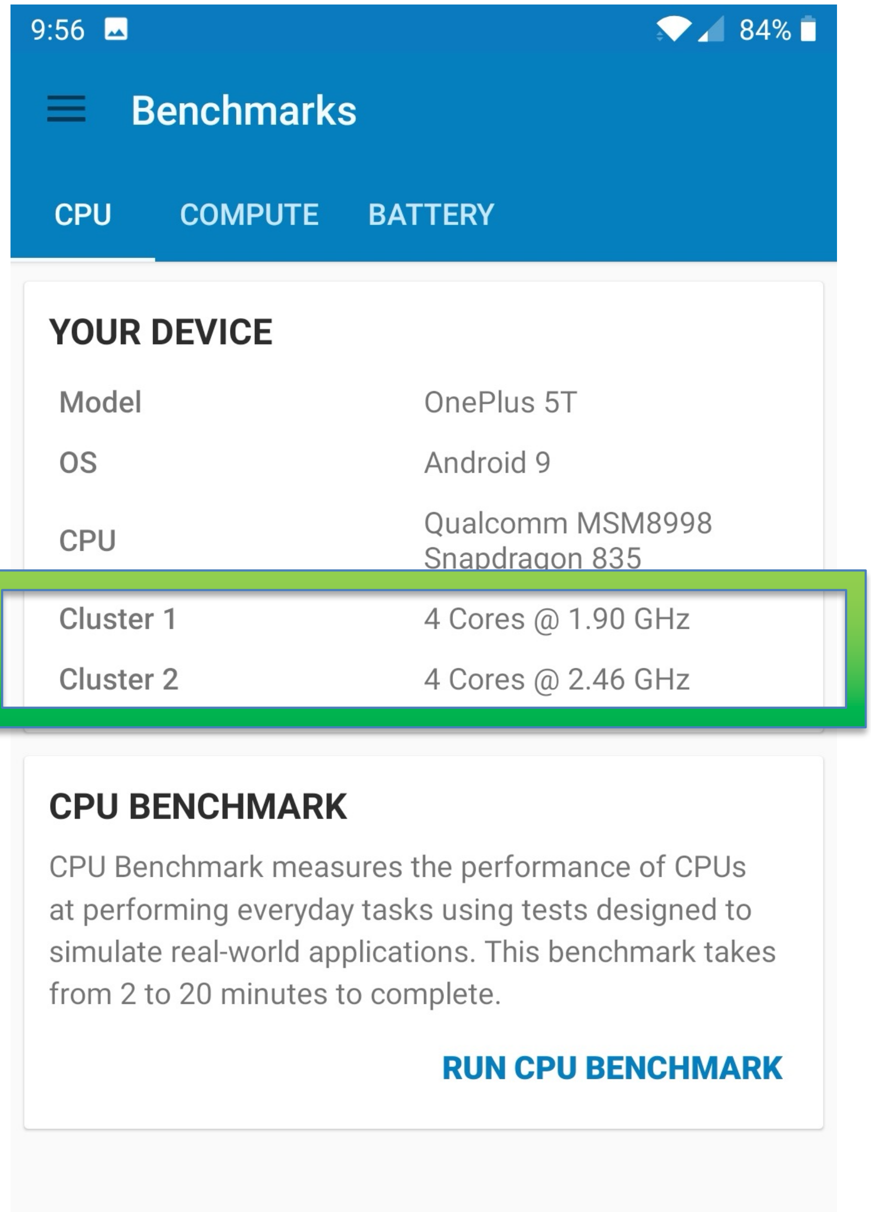
A. ≈0.707 times as long
B. ≈1 times as long
C. ≈1.207 times as long
D. ≈1.414 times as long (1/0.707)
E. None of the above

Factor by which running time changes for different programs

% of program that is parallelizable	50%	75%	90%
4 equal cores	1.25	0.88	0.65
Half-sized + 2 quarter-sized cores	1.00	0.79	0.66

Having different sized cores improves performance on less parallelizable programs at small cost on more highly parallelizable ones

Heterogeneity on a cell phone



8 cores, 2 levels of performance