Pollack's Rule

Justification for Heterogeneous Computing

Big picture

- Performance modeling: Estimating performance of a hypothetical system so system designer can compare different options

- Today: Consider different configurations of cores
  - Assumption: Total processor size (silicon area) is the same for all configurations

Pollack's rule

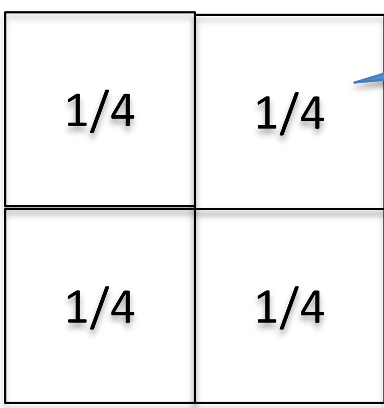
- The performance of a processing core is proportional to the square root of its area

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)

Performance: sqrt(1/4) = ½



Total performance: 4 × ½ = 2
(twice as much)

How does the running time change when a single core is replaced with 4 cores if only half the program can be parallelized?

- Parallel part:

  ½ the work / 2 the performance = ¼

How does the running time change when a single core is replaced with 4 cores if only 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

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

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 can run in paralllel

By what factor does the running time of a program that can be 75% parallelized change on 4 equal-sized cores?

Serial part: work / performance = 0.25/0.5 = 0.5

Parallel part: 0.75 / 2 = 0.375

Total time: 0.875 times as long

By what factor does the running time of a program that can be 90% parallelized change on 4 equal-sized cores?

Serial part: work / performance = 0.1/0.5 = 0.2

Parallel part: 0.90 / 2 = 0.45

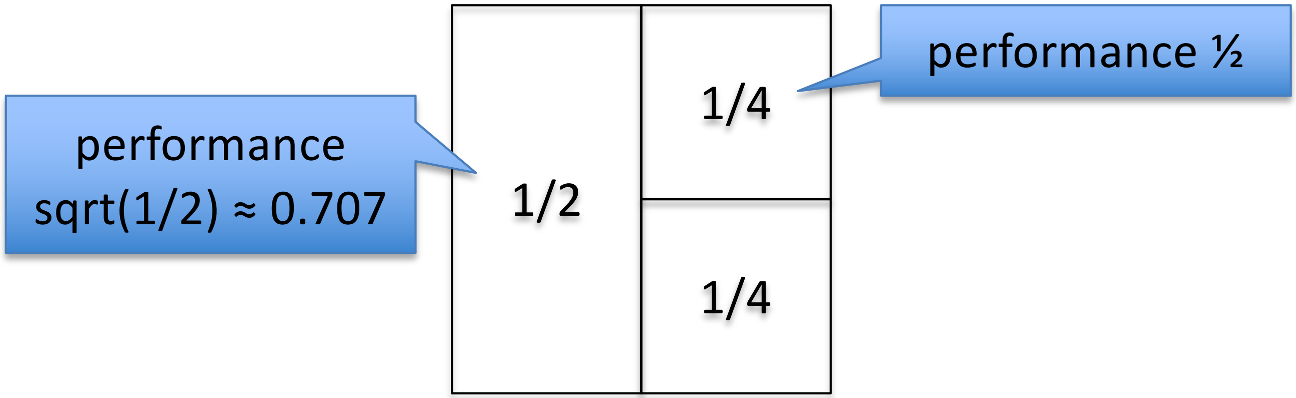
Total time: 0.65 times as long

Factor by which running time changes for different programs

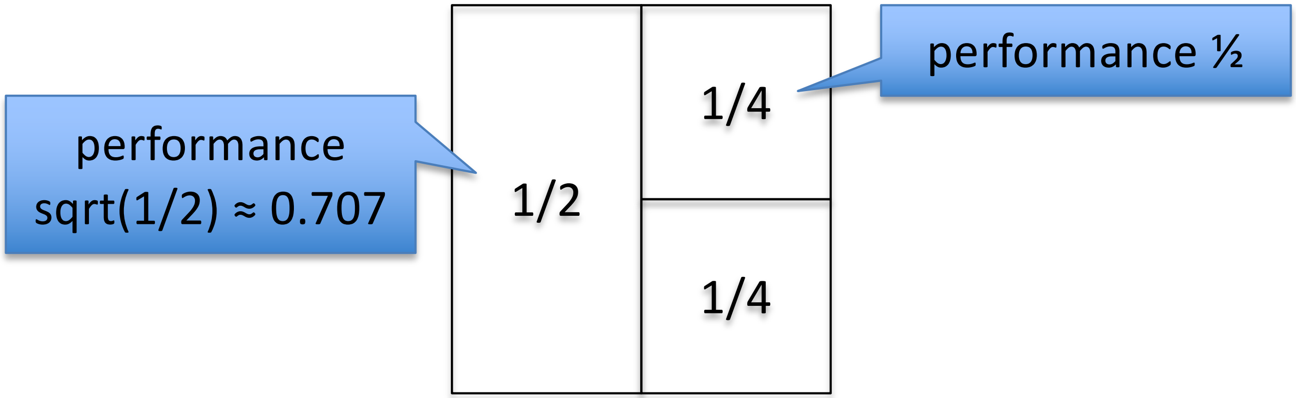
	% of program that can be parallelized		
	75%	90%	95%
1 core	1	1	1
4 cores	0.875	0.65	0.575
9 cores	1	0.6	0.467
16 cores	1.1875	0.625	0.438
25 cores	1.4	0.68	0.44
36 cores	1.625	0.75	0.458

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

What about unequal core sizes?



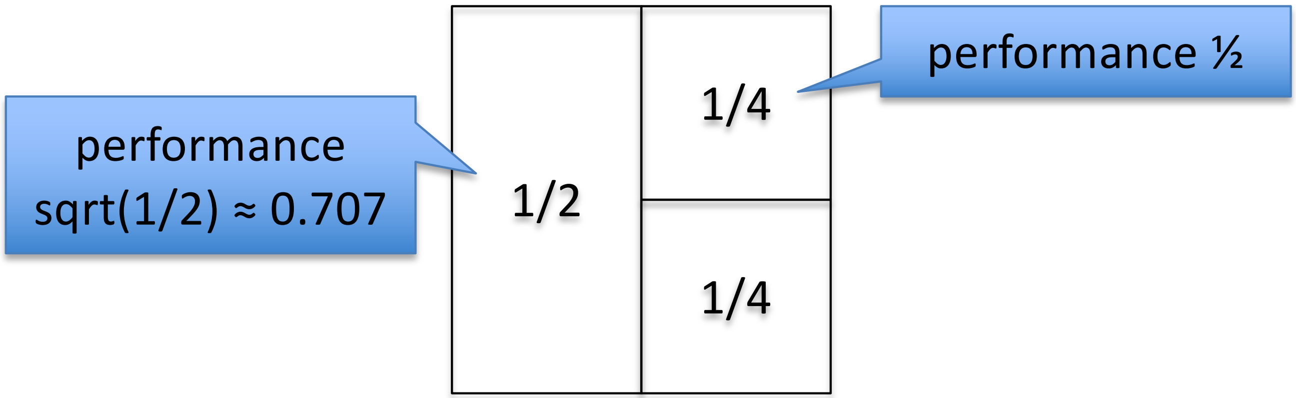
What about unequal core sizes?



By what factor does the peak performance of this system differ from a single core?

$0.707 + 2 \times 0.5 = 1.707$ times as much

What about unequal core sizes?



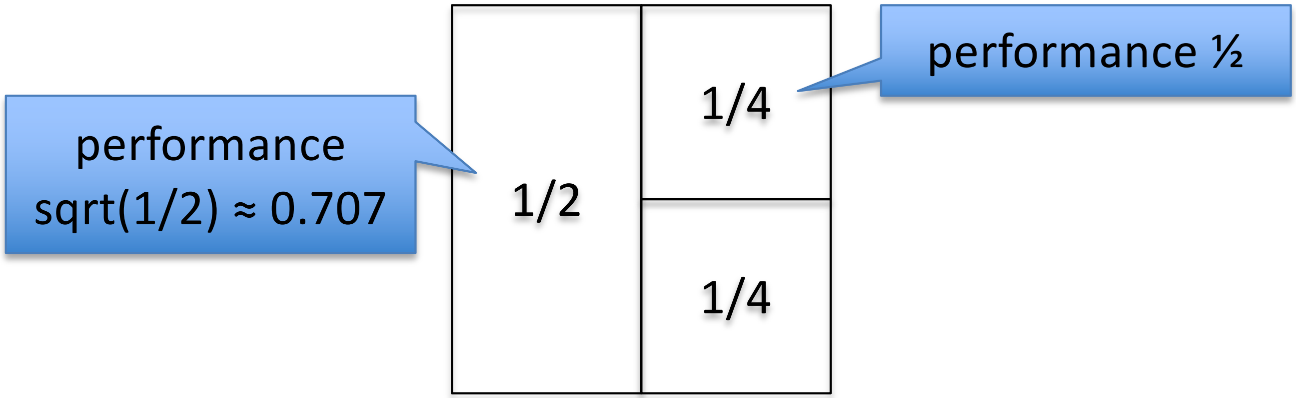
By what factor does the running time of a program that can be 75% parallelized change?

Serial part: 0.25 / 0.707 = 0.354

Parallel part: 0.75 / 1.707 = 0.439

Total time: 0.793 times as long

What about unequal core sizes?



By what factor does the running time of a program that cannot be parallelized change?

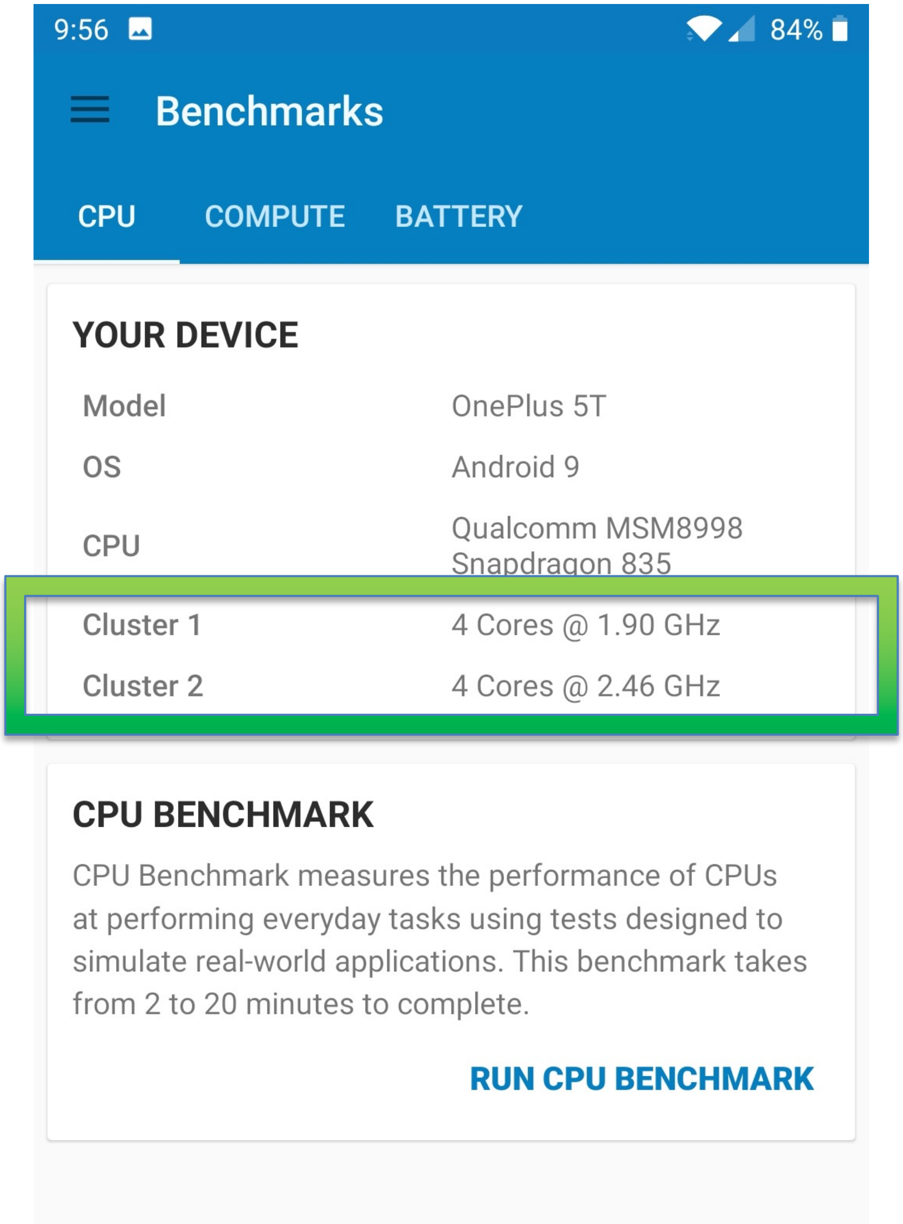
Total time = serial time = 1 / 0.707 = 1.414

Factor by which running time changes for different programs

	% of program that can be parallelized		
	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