



IBM Thomas J. Watson Research Center

# Multicore systems: Are they inherently superior?

José E. Moreira  
IBM Thomas J. Watson Research Center  
Yorktown Heights, NY

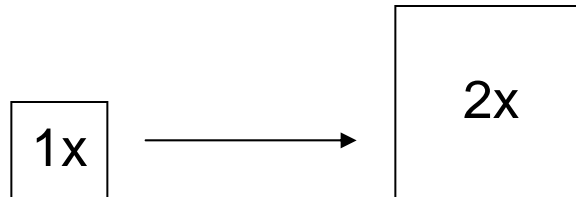


ICPADS'06 Panel

© Copyright IBM Corporation 2006

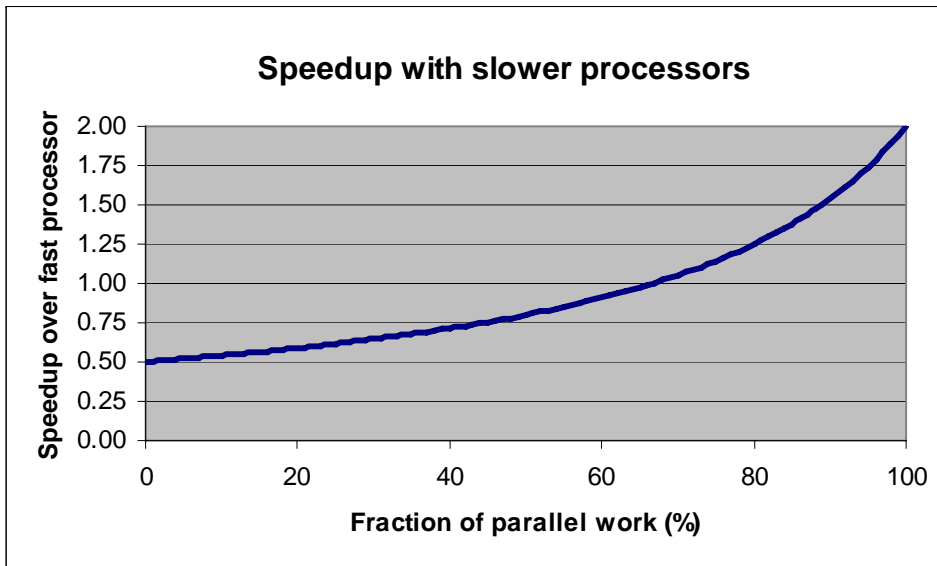
# Many slow vs. few fast processors

- Rule of thumb: A 2x improvement in single-thread performance requires 4x the area (and the power) in a given technology



- Let  $f$  be the fraction of the code that is parallel and can take advantage of more processors ( $1-f$  is the sequential fraction)
- Let  $T_1$  be the execution time on 1 fast processor
- Let  $T_4$  be the execution time on 4 slow processors =  $(f \cdot T_1 / 4 + (1-f) \cdot T_1) \cdot 2$
- Then the speedup by running the code on 4 slow processors instead of 1 fast one is  

$$S = T_1 / T_4 = 1 / (f/2 + 2 - 2f) = 2 / (4 - 3f)$$



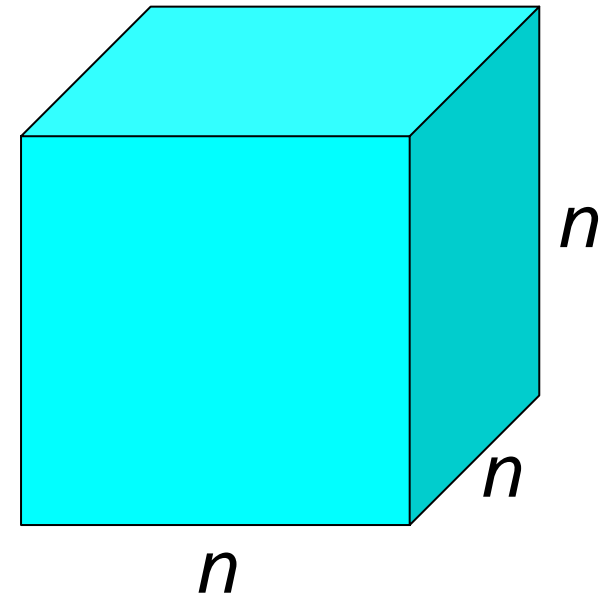
technology	# of Blue Gene cores	
	11x11mm	22x22mm
130nm	2	8
90nm	4	16
65nm	8	32
45nm	16	64

## Are there any challenges?

- The most effective use of multicore systems (and multiprocessors in general) has been as throughput accelerators
  - *The two cores in Blue Gene/L are most effective in virtual node mode*
  - *Traditional approach in commercial computing is to increase number of tasks*
- So what?
  - *Memory challenge: Not clear we can keep memory bandwidth/processor and memory capacity/processor constant!*
  - *Some kind of aggregation will be necessary!*
- What support for aggregation is available today?
  - *Scientific computing: OpenMP*
  - *Commercial computing: pthreads and Java*

## Scalability with communicating multiple cores

- Consider a parallel algorithm in which each processor operates on an  $n \times n \times n$  grid of data points
- The processors themselves are organized as a  $P \times P \times P$  grid of  $P^3$  processors
- The total problem size is  $N \times N \times N$  where  $N = nP$
- The computation time is proportional to the number of data points in each processor:  
 $T_{\text{comp}} = kn^3$
- The communication time is proportional to the surface area in each processor:  $T_{\text{comm}} = an^2 + b$



$$T_{\text{alg}} = T_{\text{comm}} + T_{\text{comp}} = kn^3 + an^2 + b = k(N/P)^3 + \underline{a(N/P)^2 + b}$$

***nonscalable!***