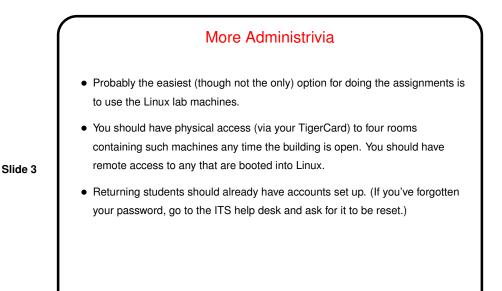
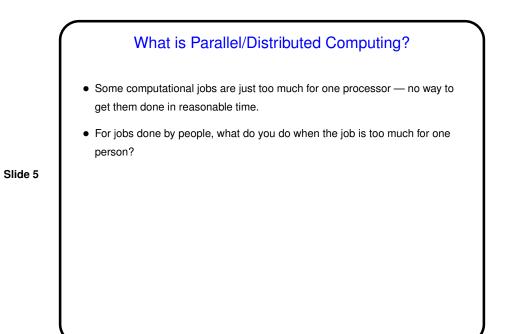


## More Administrivia Part of my job is to answer your questions outside class, so if you need help, please ask! in person or by e-mail or phone. Some of my office hours are designated as "open lab". At those times I will be in one of the labs (probably HAS 228) ready to answer questions.



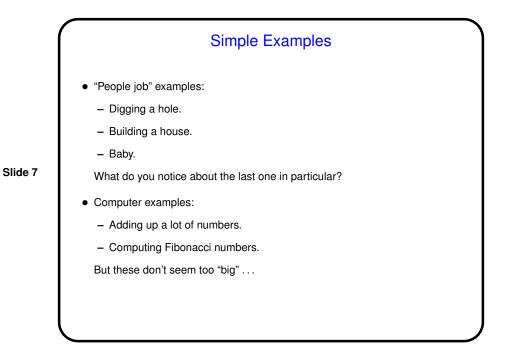
More Administrivia • Why are we using "my" book when there are books that are more textbook-like? because (1) I think it emphasizes the right things, which many textbooks don't, and (2) learning from a not-really-a-textbook and other resources should be good practice for whatever you do after you graduate. (I don't actually think I'm going to be able to retire on the extra royalty income - but it might be enough to finance a trip to Java City for the class?) Also - if you spot errors, even typos, please let me know. The first person to report any legitimate error I don't know about is eligible for extra-credit points.

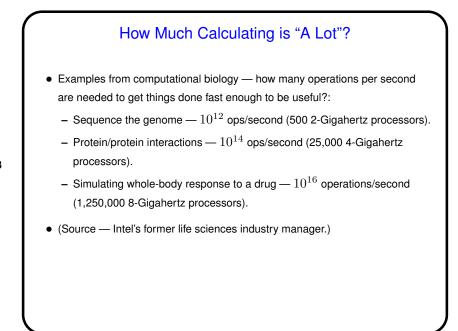


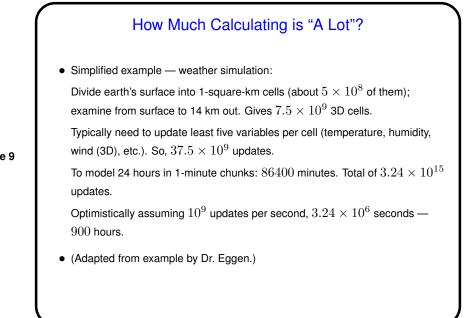
What is Parallel/Distributed Computing?
For jobs done by people, if too much for one person you assign a team — but you have to figure out

How to divide up work among team members.
How to coordinate activities of team members.

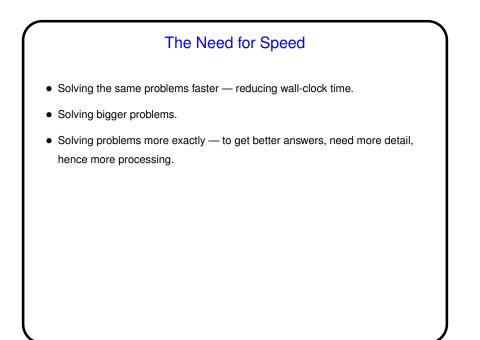
Same idea applies to computing — if too much for one processor, use multiple processors. Issues are similar — how to divide up work, how to coordinate.





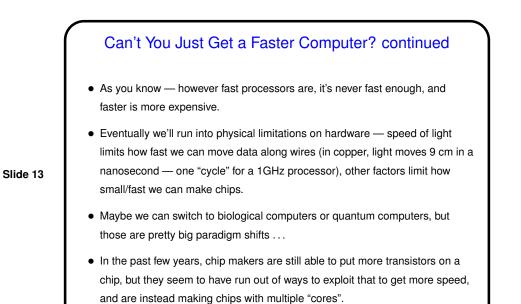


## What Are Some Other Hard Problems? Crash simulation / structural analysis. Oil exploration. Explosion simulations (why Los Alamos is interested). Astrophysics simulations (e.g., Dr. Lewis's work on Saturn's rings). Fluid dynamics. "Rendering" for computer-generated animation. And many others ...



## Can't You Just Get a Faster Computer?

- Up to a point yes. Moore's law predicts that number of transistors on a die roughly doubles every 1.5 years. Until recently, that meant doubling processor speed and memory. (Over 30 years, that's a factor of about a million!)
- But ...



"The Answer" — Parallel Computing
Analogous to "team of people" idea — if one processor isn't fast enough, use more than one.
Also useful when there's something "inherently parallel" about the problem — e.g., operating systems, GUI-based applications, etc.
http://www.top500.org tracks fastest computers; for many years now all have been "massively parallel".



- What if you aren't interested in solving problems like these "grand challenge" problems, Is there still a reason to be interested in parallel computing?
- The hardware is there, and it's becoming mainstream multicore chips, hyperthreading, etc. (The Intel person says "the chip makers can put more and more transistors on a chip, and this is the best way to use that.")
   To get best use of it for single applications, will probably need parallelism.
- Also, for some applications, thinking of them as parallel/multithreaded can lead to a solution that lets you do something useful while waiting for I/O, etc.

