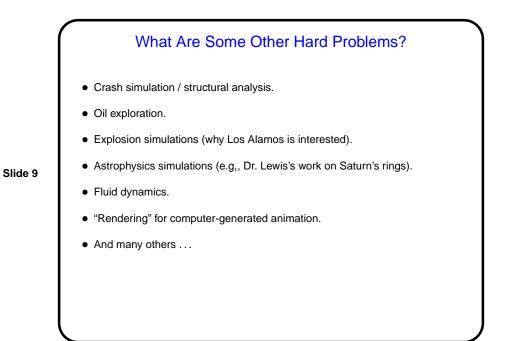
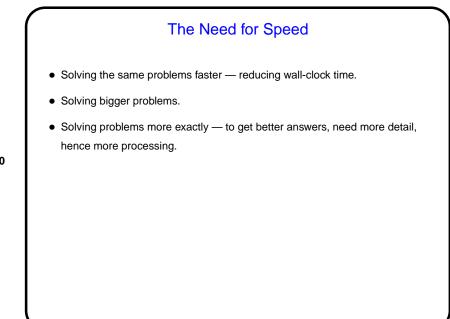


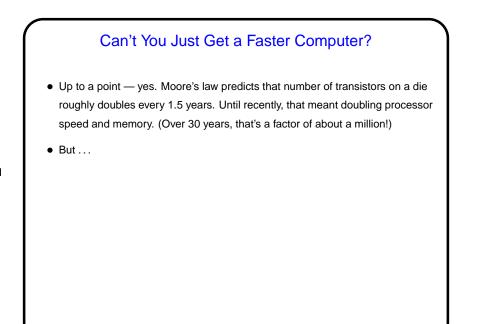
How Much Calculating is "A Lot"?
Slightly dated examples from computational biology — how many operations per second are needed to get things done fast enough to be useful?:

Sequence the genome — 10¹² ops/second (500 2-Gigahertz processors).
Protein/protein interactions — 10¹⁴ ops/second (25,000 4-Gigahertz processors).
Simulating whole-body response to a drug — 10¹⁶ operations/second (1,250,000 8-Gigahertz processors).

(Source — Intel's former life sciences industry manager.)

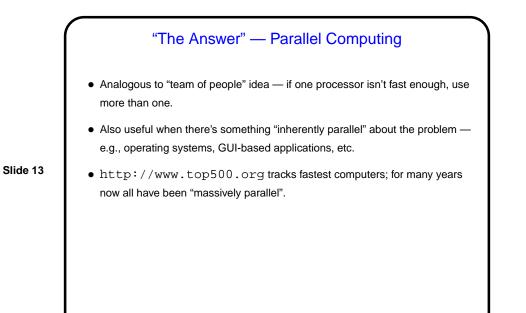






Can't You Just Get a Faster Computer? continued

- As you know however fast processors are, it's never fast enough, and faster is more expensive.
- Eventually we'll run into physical limitations on hardware speed of light limits how fast we can move data along wires (in copper, light moves 9 cm in a nanosecond — one "cycle" for a 1GHz processor), other factors limit how small/fast we can make chips.
- Maybe we can switch to biological computers or quantum computers, but those are pretty big paradigm shifts ...
- In the past few years, chip makers are still able to put more transistors on a chip, but they seem to have run out of ways to exploit that to get more speed, and are instead making chips with multiple processing elements ("cores" for computational chips, other elements in GPUs).



But I Don't Want To Solve Problems Like Those!
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, general-purpose computing GPUs, 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.

