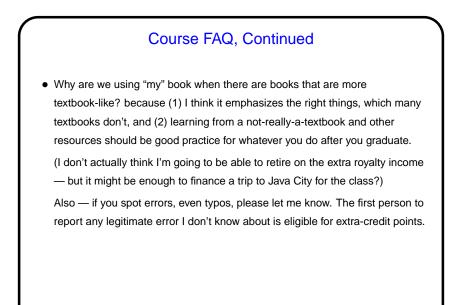
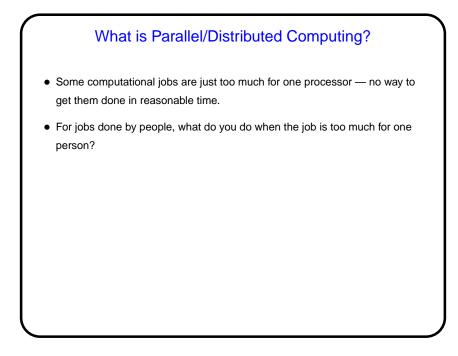
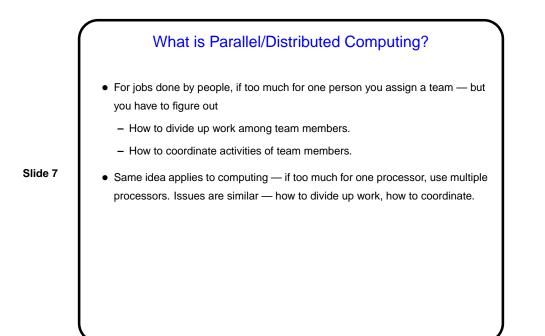
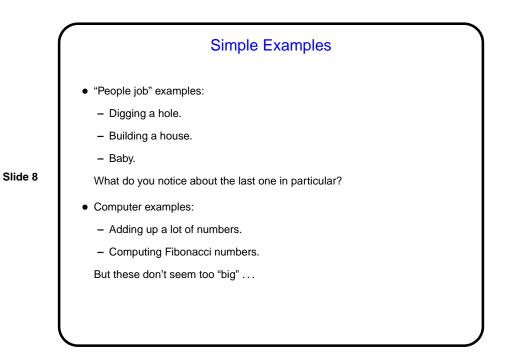


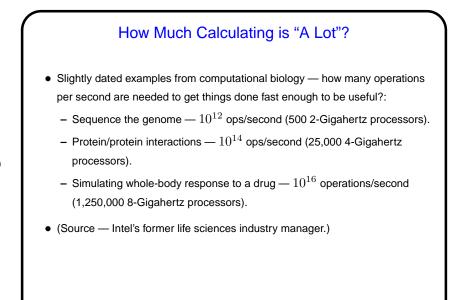
Course FAQ, Continued • "What computer(s) can I use to do homework?" Easiest option is probably department's Linux lab machines. There are others. You should have physical access (via your TigerCard) to four rooms containing such machines any time the building is open. You should have remote access to any that are booted into Linux. Returning students should already have accounts set up. (If you've forgotten your password, go to the ITS help desk and ask for it to be reset — but be sure they know it's for the CSCI/ENGR system.)



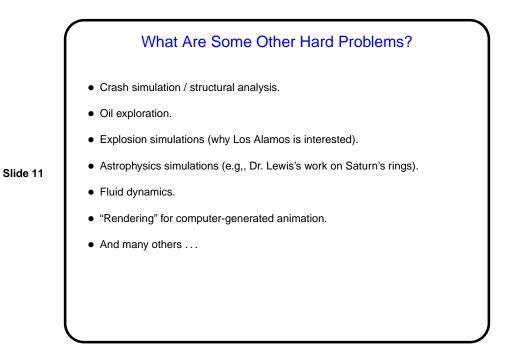


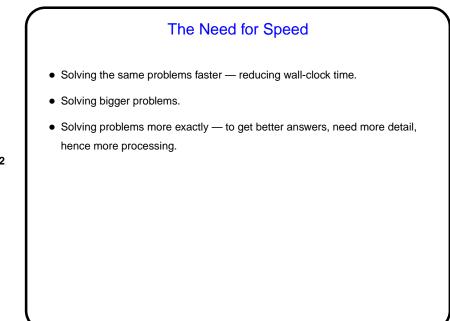


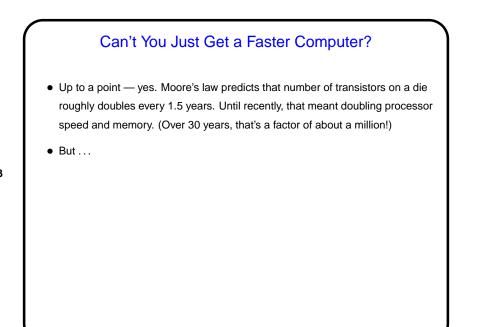




How Much Calculating is "A Lot"?• Another simplified example — weather simulation:
Divide earth's surface into 1-square-km cells (about 5×10^8 of them);
examine from surface to 14 km out. Gives 7.5×10^9 3D cells.
Typically need to update least five variables per cell (temperature, humidity,
wind (3D), etc.). So, 37.5×10^9 updates.
To model 24 hours in 1-minute chunks: 86400 minutes. Total of 3.24×10^{15}
updates.
Optimistically assuming 10^9 updates per second, 3.24×10^6 seconds —
900 hours.• (Adapted from example by Dr. Eggen.)

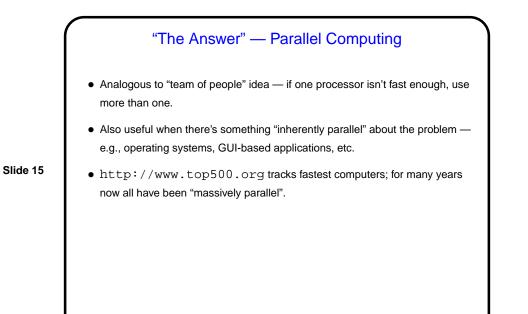




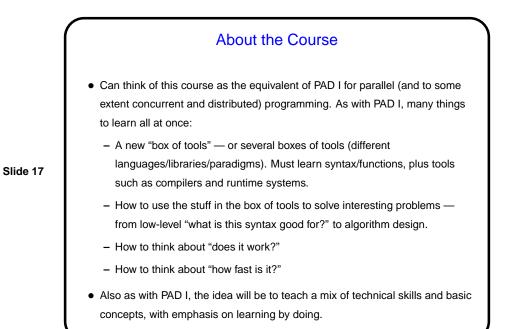


Can't You Just Get a Faster Computer? continued

- 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 "cores".



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, 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.



Minute Essay
What are your goals for this course?
Are you reasonably comfortable with Java and C? How about C++? (There will be assignments using both C and Java.)
Do you have any experience already with parallel or multithreaded programming? If so, tell me about it, briefly.
Anything else you want to tell me? about the course, about what you did over the break ...