11:59:07 From devin to Everyone: When will grades for Project 0 come out?

TAs are working on it now. There are 490 to do.

12:04:21 From Ben Wichser to Everyone: Won’t it be the number of trials, not the number of cores, that can alter the probability? Since the tries are all over the same data set?

Correct. NUMTRIALS affects the probability, at least up to a certain point. NUMT does not.

12:05:05 From Tremper, Brayden Justin to Everyone: 90% and up?
12:05:12 From Nord, Casey William to Everyone: I'm consistently getting 0.97
12:05:37 From Ricardo Wu to Everyone: I am getting 99% on my testruns as well
12:06:03 From Wil Coiner to Everyone: .94 here because MacOS hates OpenMP lol
12:06:29 From Alcaide, Tiffanie Charlyne Yu to Everyone: So in general for projects, there’s not necessarily a “right or wrong” answer? Because of the differences in everyone’s systems?

That’s pretty much true. Also, there is a difference in load average. Generally when we grade, there is a range of results that we will accept. No, I don’t have that written down anywhere.

12:32:49 From Matsumoto, Nicholas to Everyone: Do you have to multiply the area at the end [of Project #2] by two?

It’s volume, not area. But, yes, we are only integrating the top half, but I want the volume for both top and bottom halves together. Since they are symmetric, just multiply by 2.

12:33:17 From alexa to Everyone: In main, I see … and ??
Do those both refer to code we need to fill in ourselves)?

Yes.

12:33:37 From Matsumoto, Nicholas to Everyone: For the modulus and division way of iterating over the array is it possible to make that work on arrays with non-matching columns and rows?

Yes, you mod and divide by the number of columns.

12:33:38 From Taylor, James Edward to Everyone: Wait then why don’t we just do one corner and multiply by 8?

You can – this will work fine if you work through the logic correctly.

12:34:02 From Abhi Balijepalli to Everyone: What does the 3 dots mean?

The 3 dots and the question marks all refer to things you need to fill in.

12:35:04 From Cho, Yongsung to Everyone: Don't we have to modify the Height function?

No, it is meant to be used as-is.
12:48:52 From Wil Coiner to Everyone: So [performance] array > linked list on modern machines because caching?

I believe so. Also, there is a little less computation to getting to the next array entry than there is to following a link.

13:02:57 From Patrick Sullivan to Everyone: How are addresses determined to be on the same cache line? Is it just a mask of the address to determine what gets pulled into the cache?

Absolutely right! Yes, it’s a mask. Since the cache line is 64 bytes, and 64 is represented in 6 bits, then two memory addresses are in the same cache line if their top 26 (32-6) or 58 (64-6) bits are the same. Cache lines in memory start at addresses where the bottom 6 bits are all zeroes.

13:05:26 From James Taylor to Everyone: Creel has a good video explaining how caches work. https://www.youtube.com/watch?v=UCK-0fCchmY

Nice!

13:28:58 From James Taylor to Everyone: As a matter of fact some compilers do this.

Usually the reason compilers pad out structs is to keep multi-byte types from starting on illegal boundaries. For instance, a struct like this:

```c
struct ci {
    char c;
    int i;
};
```

would become:

```c
struct ci {
    char c;
    char pad[3];
    int i;
};
```

or, `char pad[7]` if using 64-bit ints.

so that `i` will start on a multiple-of-{4,8}-byte boundary.

13:29:08 From Cruz Solano to Everyone: is industry pushing towards solving this false sharing problem within the chip rather doing it through software?

No, as far as I know.

13:34:00 From Patrick Sullivan to Everyone: Is there a command that shows cache hit/miss counts pcpu?

I think some of the Intel performance profiling and tuning tools (vtune?) do that.
13:35:46 From Burke, Caden Thomas to Everyone: So it's not essentially a line, just a group in memory corresponding to the mask

It's always called a cache line, but, you're right, it is not a line but a 64-byte chunk of memory.

13:40:16 From Patrick Sullivan to Everyone: So L1 cache is per core but L2 is per processor?

Both L1 and L2 caches are per-core. L3 is per-CPU.

13:41:14 From Patrick Sullivan to Everyone: Do these cache invalidation issues come up for L1 as well or was L2 just the example?

They would, but the L1 caches are pulling from their core's L2 cache, not from memory.

13:41:50 From Headrick, David Joshua to Everyone: Can adjusting chunksize in aomp parallel for loop help with false sharing?

It could. If you adjust chunksize so that the data for one core's for-loop passes fall on different cache lines than the data for any other core's for-loop passes, then there won't be any False Sharing because no two cores will use the same cache lines.

14:00:33 From alexa to Everyone: Are iv and iu the length and width of the superquadric or some part within it? As in, are these tiles within tiles, or are we just going to create more tiles by adding more dots?

You can create more tiles by adding more dots. In the program, NUMNODES is the number of dots on each edge of the grid. Thus, there are a total of NUMNODES^2 dots. iu indexes dots from left to right. iv indexes dots from bottom to top. Both iu and iv go from 0 to NUMNODES-1.

14:02:15 From Huy Trieu to Everyone: Should NUMNODES be defined as a global variable?

The Project Notes talk about there being two approaches to getting the independent variables into your program (slides #18-30). In Approach #1, NUMNODES and NUMT will be #defines. In Approach #2, NUMNODES and NUMT will be globals.

14:11:49 From alexa to Everyone: The parallelization aspect is--dividing the bottom area of the superquadric into a certain number of quadrants and computing the heights of the tiles within the quadrant in separate threads? (Just want to make sure I understand what exactly we're parallelizing)

Yes. NUMNODES^2 is the "data set size", which you are parallelizing over.

15:07:39 From Arvin to Everyone: For project 1, it says we need to test 1, 2, and 4 threads but more are OK. If we did 8 threads as well, should we compute Fp using the results from 4 or 8 threads or either is fine?

I would use the run that had the largest number of threads and the largest data set size to compute Fp.

15:14:24 From Huy Trieu to Everyone: how large should NUMNODES be?

Try some small values and some big ones, maybe, 3, 10, 50, 100, 500, 1000

15:58:57 From Lucian's PC to Everyone: I'm running project 2 and my results seem to converge as the NUMNODES increases, but not NUMT.

That's what you should see. NUMT does not affect the computation, just the performance.