

OpenMP (cont.)

Lecture 11
April 8, 2025

To Dos

Reading for next time (GPUs!)

Extra credit

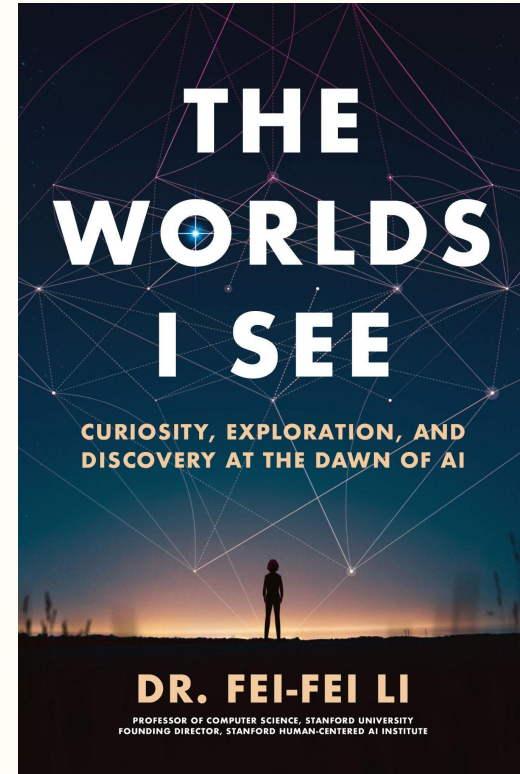
Program 4 presentations and
discussion

Extra Credit

30 Minute Discussion on Book

In May before end of semester

~ step up on final grade (e.g., A- to A)



Some More Synchronization Directives

- `#pragma omp critical (name)`
 - Creates critical section around structured block of code
 - Use `name` to enable multiple, non-interfering critical sections
- `#pragma omp barrier`
 - Block until all threads in team reach this code, then all proceed
- `#pragma omp atomic`
 - High performance critical section
 - Single C statement of form `auto-inc/dec` or `x<op>=expression`

Locks

- `void omp_init_lock(omp_lock_t *lock_p)`
 - Initialize to unlocked
- `void omp_set_lock(omp_lock_t *lock_p)`
 - Grab the lock
- `void omp_unset_lock(omp_lock_t *lock_p)`
 - Release the lock
- `void omp_destroy_lock(omp_lock_t *lock_p)`
 - Uninitialize the lock

Tasking

- Enables parallelism when unbounded number of loop iterations
- Allows specification of independent computation
- `#pragma omp task`
 - Generates new task that will be scheduled for execution
 - Can be used within `parallel` directive, but often with `single` directive
- `#pragma omp task if(n > 2)`
 - Only creates task if `n > 20`