Programming Assignment #4: Merge Sort with a Thread Pool

Prof. Li-Pin Chang

National Chiao-Tung University

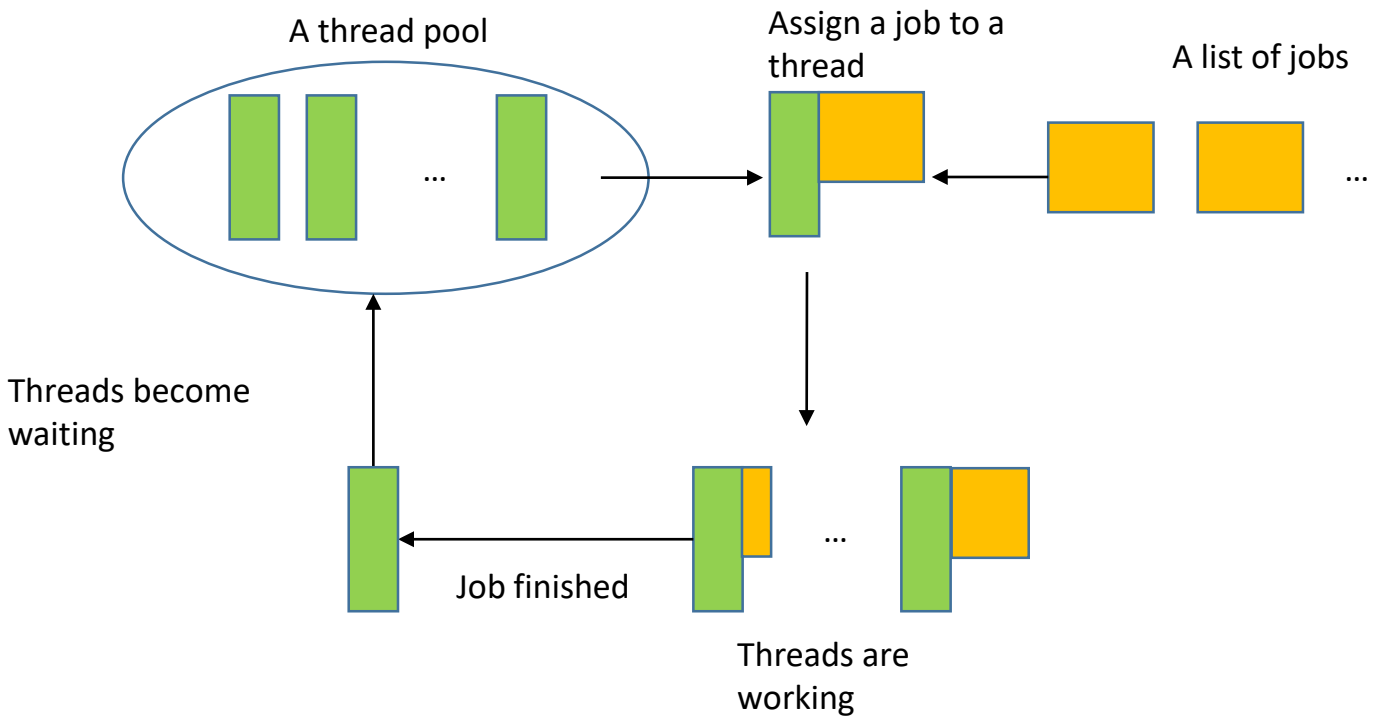
Objective

- Multithreaded sorting using a thread pool
  - # of threads in the pool determines the max. degree of parallelism
- The problem definition is the same as that in the previous assignment, except that the binding of jobs to threads is dynamic

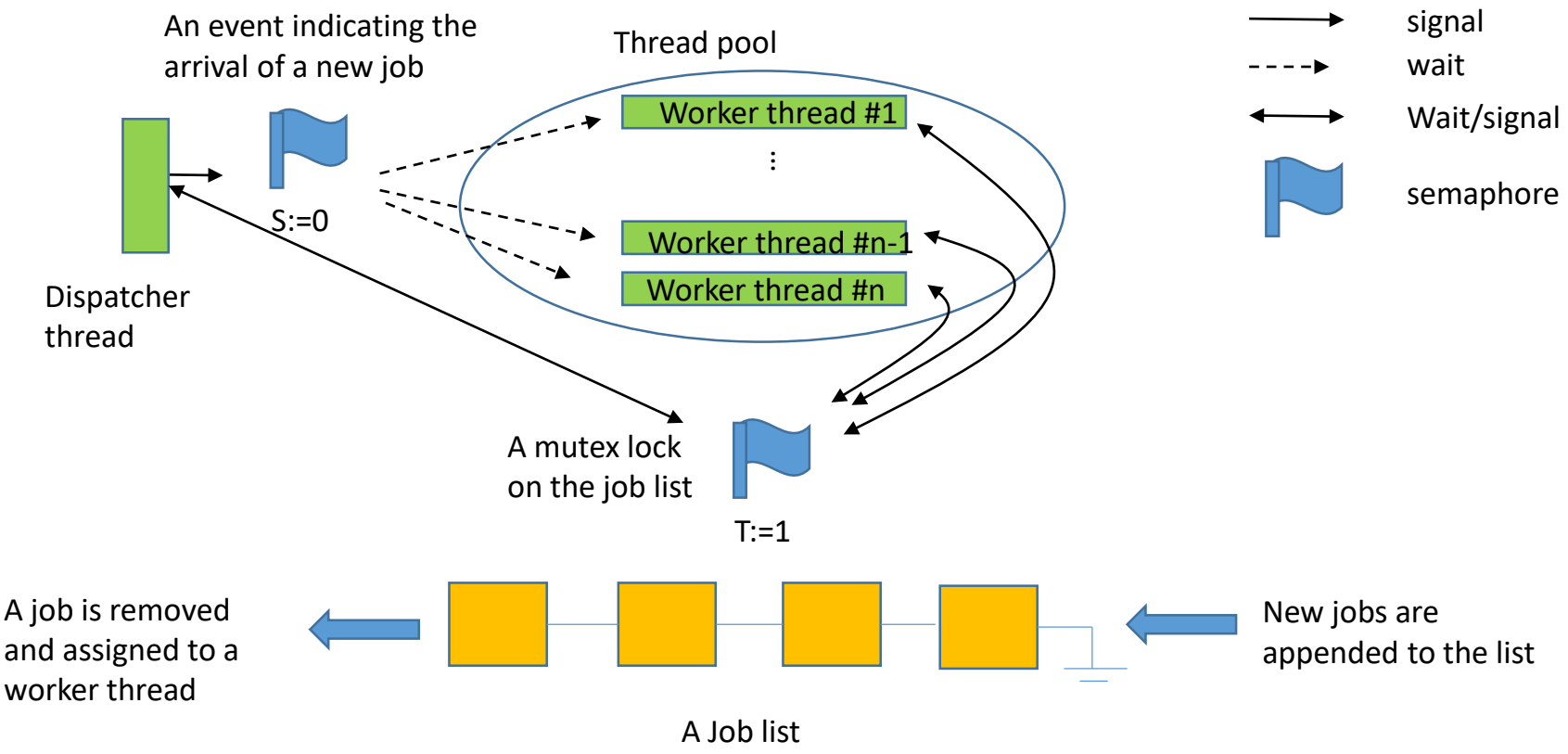
Job Binding

- A job is
  - Sorting a last-level array using bubble sort, or
  - Merging two sorted sub-arrays into a large, sorted array
- All threads in the thread pool must have been created before the first job starts

The Concept of a Thread Pool



A Reference Design



A Reference Design

- Initially, the dispatcher thread inserts eight sort jobs for the eight bottom-level arrays and wakes up worker threads
- The dispatcher checks if any two pairing (buddy) sub-arrays have been sorted. If so, it inserts a new job to merge the two sub-arrays

Procedure

1. Read data from the input file “input.txt”
2. n=1
3. Do the sorting with a thread pool of n threads
4. Print the execution time
5. Write the sorted array to a file
   - Filename: output_n.txt (e.g., output_3.txt if n=3)
6. n++; if n<=8 then goto 3

Remarks

- Reuse your assignment 3
- The binding of jobs to threads must be dynamic
- All the 8 output files must be identical
- Execution time decreases as *n* increases
- Performance improvement saturates as *n* increases
- ***You got 0 mark if you use quicksort***

Input/Output Format

- Format of “input.txt”:

<# of elements of array><space>\n

<all elements separated by space>

  - Largest input: the same as in assignment #3

- Output file format “output_?.txt”:

<sorted array elements separated by space>

Header of your .c or .cpp

/*

Student No.: <your student id>

Student Name: <your name>

Email: <your email>

SE tag: xnxcxtxuxoxsx

Statement: I am fully aware that this program is not supposed to be posted to a public server, such as a public GitHub repository or a public web page.

*/

Turn in your program

- <student id>_<assignment #>.c or .cpp
- E.g., 0756073_1.c or 0756073_1.cpp

Testing OS Environment

- Ubuntu 16.04, Ubuntu 14.04 or CS linux work station
  - Your code should compile successfully in one of the above environments