A speedup vs. the number of workers plot is a graphical representation that illustrates how the speedup of a parallel algorithm or program changes as you increase the number of worker (processing) units. It is a common way to assess the efficiency of parallel computing and to determine how well a parallel algorithm scales as more resources are added.

In a speedup plot:

- The x-axis typically represents the number of worker units, which can be processors, cores, threads, or any parallel execution units available in a computing system.

- The y-axis represents the speedup achieved by the parallel algorithm. Speedup is a measure of how much faster a parallel program runs compared to its sequential (single-threaded) counterpart. It is usually calculated as:
    - **Speedup = Sequential Execution Time / Parallel Execution Time**

- Sequential Execution Time: The time it takes for the program to run on a single worker unit (e.g., a single processor or core).

- Parallel Execution Time: The time it takes for the program to run on multiple worker units.

In a speedup vs. the number of workers plot:

- As you increase the number of worker units (x-axis), you measure the corresponding speedup (y-axis) achieved by the parallel program.

- Ideally, as you add more worker units, the speedup should increase linearly, showing that the program scales efficiently with more resources.

- A perfect linear speedup (a 1:1 ratio) would indicate that the program is using the additional resources effectively, and the execution time decreases linearly as you add more workers.

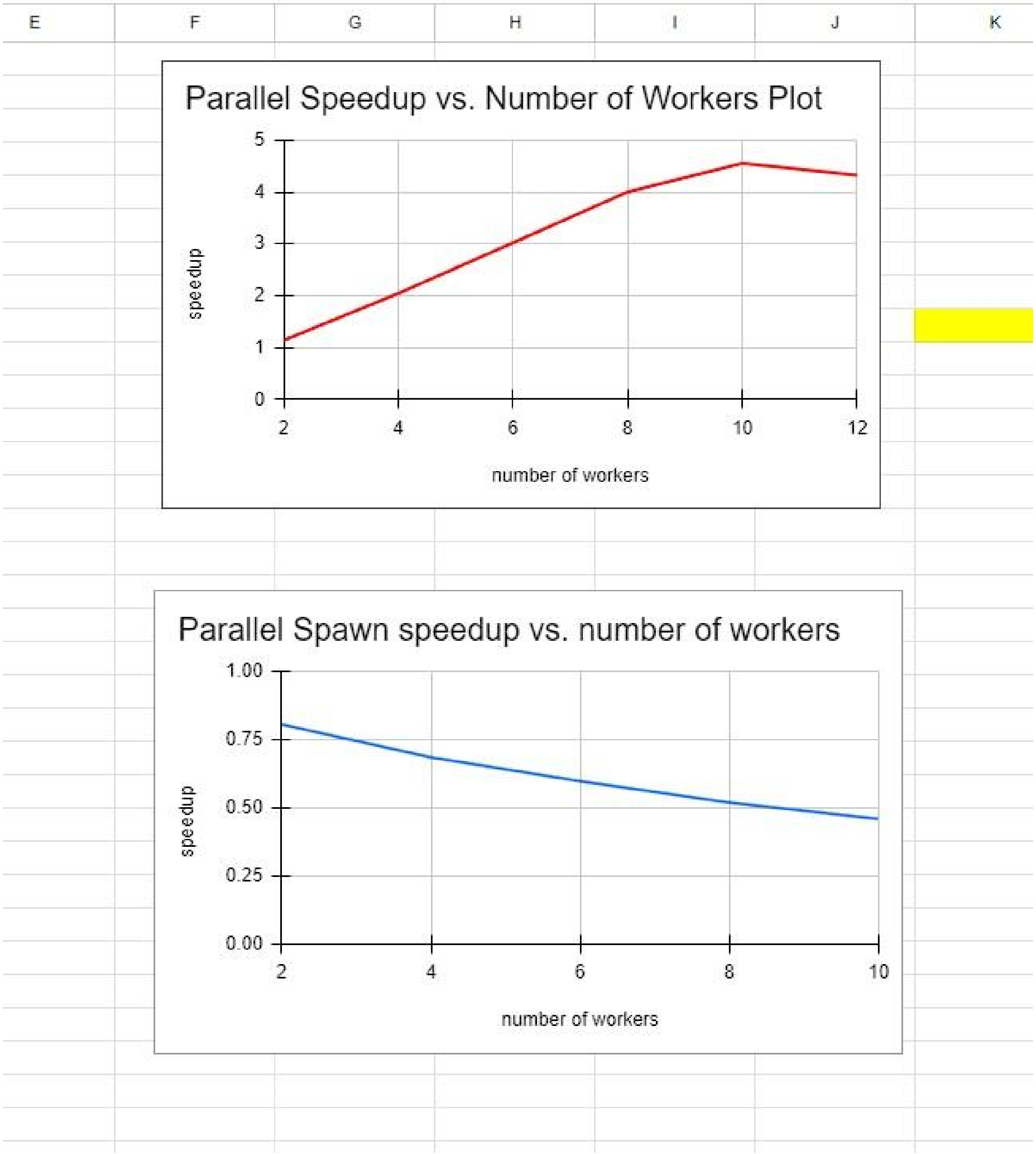
- In reality, achieving perfect linear speedup is often challenging due to factors like communication overhead, synchronization, and load balancing. Therefore, speedup curves may eventually level off, showing diminishing returns as you add more workers.

- A superlinear speedup (speedup greater than the number of worker units) may occur in some cases due to various factors, such as better cache utilization, reduced contention, or algorithmic optimizations.

- If the speedup curve starts to flatten out or even decrease as you add more workers, it indicates diminishing returns or potential inefficiencies in the parallelization.

Analyzing speedup vs. the number of workers plots helps us understand the scalability of a parallel program and make decisions about the optimal number of worker units to use for a given problem size. It can also highlight potential bottlenecks or areas for improvement in the parallel algorithm or implementation.

The plot I got was:


E
F
G
H
I
J
K
Parallel Speedup vs. Number of Workers Plot
speedup
number of workers
Parallel Spawn speedup vs. number of workers
speedup
number of workers