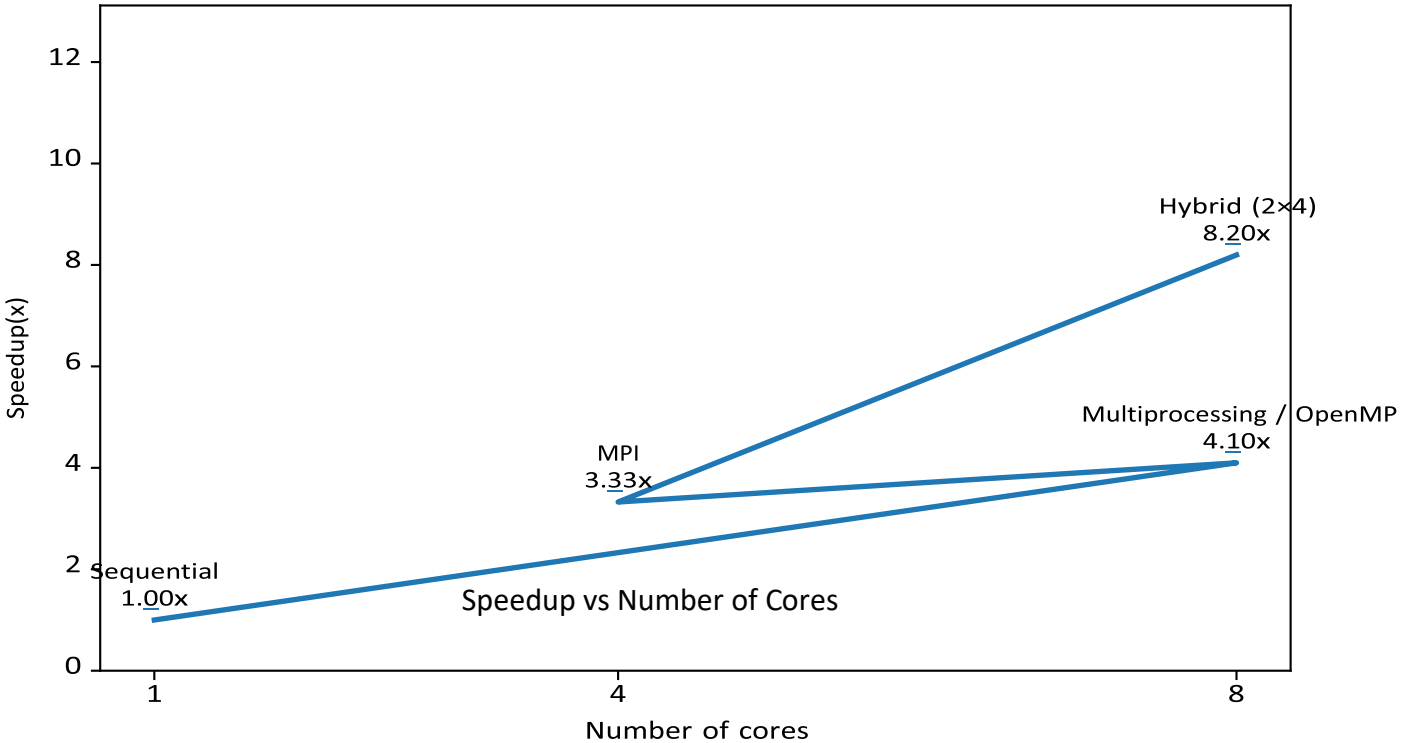
12
10
8
6
4
2
0
Speedup(x)
Hybrid (2×4)
8.20x
Multiprocessing / OpenMP
4.10x
MPI
3.33x
Sequential
1.00x
Speedup vs Number of Cores
1
4
8
Number of cores

Method	Cores / Nodes	Time (s)	Speedup	Efficiency
Sequential	1	58.2	1.00x	100.0%
multiprocessing /OpenMP	8	14.2	4.10x	51.2%
MPI	4	17.5	3.33x	83.1%
Hybrid (2×4)	8 (2×4)	7.1	8.20x	102.5%

Discussion:

- Efficiency drops beyond a certain point because overheads (synchronization, memory contention, and communication) grow while useful work per core decreases.
- Communication overhead (MPI scatter/gather and synchronization) reduces the parallel fraction, especially when per-node work is small and message latency dominates.
- Workload imbalance (e.g., one node getting extra rows) causes faster nodes to idle waiting for the slowest node, lowering overall throughput and efficiency.
- Hybrid (MPI + intra-node multiprocessing) was most scalable here because it reduced inter-node communication while exploiting shared-memory parallelism within nodes.

To maximize performance tune local worker count to match physical cores and avoid oversubscription.

-