What is Parallel Computing

serialized task를 parallel task로 decomposition 하고
computer source를 각 task에 mapping 하여
simultaneous operation을 통해 performance를 향상시키는 것

Why parallel computing

It can reduce execution time
It can efficiently use computer resource (without idle)

Open MP

: open specification for Multi Processing
- execution model : fork-join
- memory model : shared memory

false sharing

→ same cache line 공유
write on it .. sync 되기 때문에 performance 저하되는 것

Shared memory system에서 발생하는 performance reduction의 요인
when multiple thread occupy different core,
It shares L3 cache and data is transfered in cache line unit.
When each thread access contiguous memory region,
Each element or chunk is smaller than cache line,
They share same cache line.
Then when they copy whole cache line which contains other elements.
If they do write operation on it,
Data inconsistency / cache coherence violated. so
It should synchronize ● cache line. validate/update occurs.

Parallel Algorithm design

① two major steps → decomposition / Mapping

② critical path is longest weighted path
It represents lower bound / shortest time on that parallel algorithm

③

④ Task Interaction graph
: data interaction이 중요함. crossing edge = communication cost

⑤ recursive decomposition
- when find minimum, we can decompute one serialized task into several chunks

⑥ Data decomposition
- Input / Input-output / Output / Intermediate data를 decomposition 하는거.
- data만 나누고 computation 과정은 똑같기 때문에 가능

⑦ general design guide for minimizing interaction
① graph based static mapping, ~~output~~ crossing edge 최소로 가도록
② copy를 통해 interaction 줄이기
③ communicate bandwidth 이용하도록 Volume↑ frequency↓
④ overlapping computation — communication