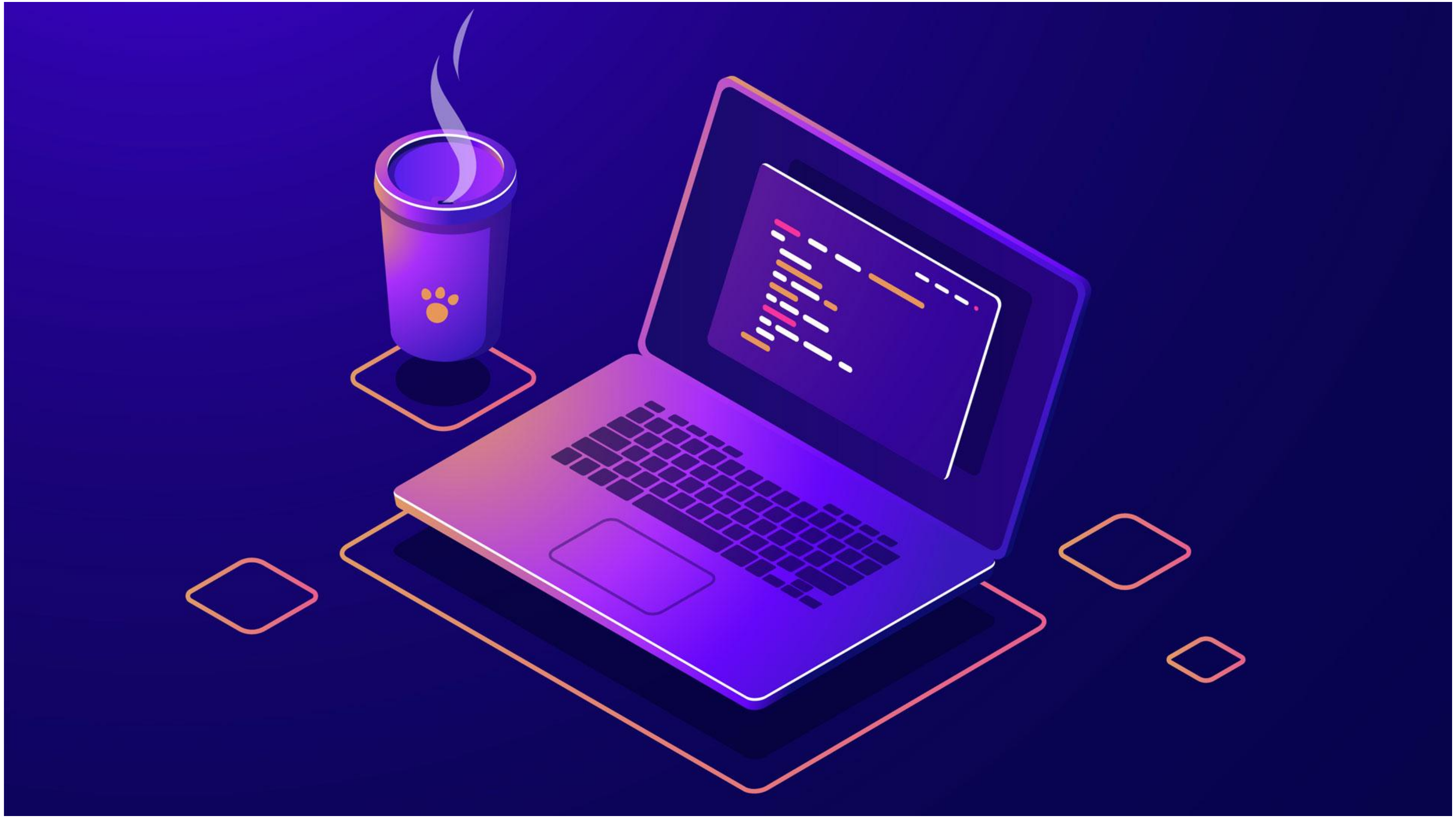
Common Overheads

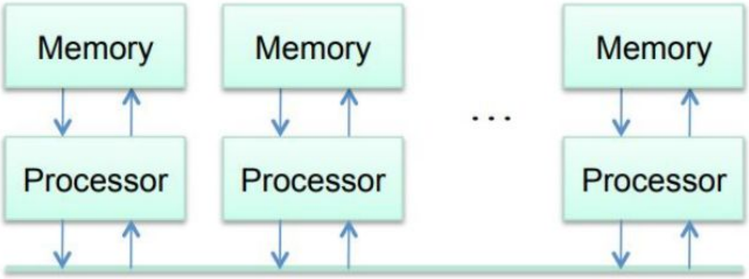


Common Overheads

- **Data sharing:** can be explicit via messages, or implicit via a memory hierarchy (caches)
- **Idleness:** thread cannot find any useful work to execute (e.g. dependences, load imbalance, poor communication and computation overlap or hiding of memory latencies, ...)
- **Computation:** extra work added to obtain a parallel algorithm (e.g. replication)
- **Memory:** extra memory used to obtain a parallel algorithm (e.g. impact on memory hierarchy, ...)
- **Contention:** competition for the access to shared resources (e.g. memory, network)

How to model data sharing overhead?

We start with a simple architectural model in which each processor P_i has its own memory, interconnected with the other processors through an interconnection network.



- Processors access to local data (in its own memory) using regular load/store instructions
- We will assume that local accesses take zero overhead.

How to model data sharing overhead?

- Processors can access remote data (in other processors) using a message-passing model (remote load instruction[2])
- To model the time needed to access remote data we will use two components:
  - Start up: time spent in preparing the remote access (t_s)
  - Transfer: time spent in transferring the message (number of bytes m, time per byte t_w) from/to the remote location

$$t_{access} = t_s + m \times t_w$$

- Synchronization between the two processors involved may be necessary to guarantee that the data is available

How to model data sharing overhead?

Assumptions (to make simpler the model)

- At a given moment, a processor P_i can only perform one remote memory access to another processor P_j
- At a given moment, a processor P_i can only serve one remote memory access from another processor P_k
- Both actions can be performed simultaneously: $P_i \rightarrow P_j$ and $P_i \leftarrow P_k$

Instructor Social Media

Youtube: Lucas Science

Instagram: lucaasbazilio

Twitter: lucasebazilio