

Logical Clocks

Definition

Logical clocks are constantly updated timestamps that enable a basic partial ordering of events, in local as well as distributed systems.

Lamport clocks

- ① Each processor/core P_i has a logical clock LC_i
- ② An event occurs on processor $P_i \Rightarrow LC_i = LC_i + 1$
- ③ When a processor P_x sends a message to P_y , if LC_x also sends its logical clock LC_x
- ④ P_y receives the message \Rightarrow if $(LC_y > (LC_x + 1))$ else $LC_y = LC_x + 1$

Vector clocks

- A vector clock is similar to the Lamport clock above, but each process keeps track of the clock of each other process. It is in essence n Lamport clocks for each process, where $n = \text{number of processes}$.
- When a process receives a message, it merges its clock-vector with the clock-vector in the message, finding the maximum of each item.
- Vector clocks overcome the shortcoming of Lamport clocks which is $L(e) < L(e')$ does not imply e happened before e' .
- Vector clocks capture causality, which Lamport clocks do not.
- However, Vector clocks are more expensive in terms of bandwidth.