EE 382N: Distributed Systems **Fall 2017**

Lecture 1: September 5

Lecturer: Dr. Vijay Garg *Scribe: Shailesh Kelkar and Daniel Cabrera*

1.1 Recap

Happened-before relation ($\rightarrow$) is the smallest relation such that
1. if ($e \prec f$) in the same process, then (e$\rightarrow$f) where $\prec$ is locally precedes.
2. if ($e \rightsquigarrow f$), then (e$\rightarrow$f) where $\rightsquigarrow$ is remotely precedes.
3. if ($\exists g$:($e \rightarrow g$) $\wedge$ ($g \rightarrow f$)), then (e$\rightarrow$f).

Thus, $\rightarrow$: $(\prec \cup \rightsquigarrow)^+$ where + is transitive closure.

1.2 Topics

1. Logical Clock
2. Physical Clock
3. Down-sets
4. Principal Ideals
5. Vector Clock
6. Dilworth's Theorem

1.3 Logical Clock

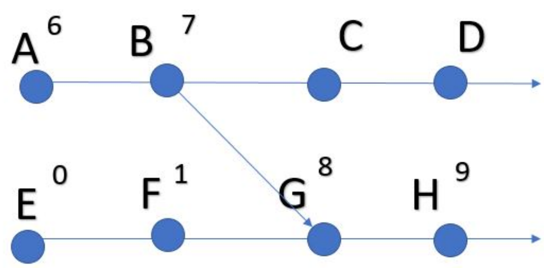
1.3.1 Definition

A map C:E $\rightarrow \mathbb{N}$ is a logical clock if $\forall$ e,f $\in$ E: ($e \rightarrow f$) $\Rightarrow$ C(e) $<$ C(f). In short, C is the function that preserves the structure.

1.3.2 Logical Clock Algorithm

The variable c (integer) is initialized to 0.
1. For internal events, c++ (increment c by 1).
2. For send event, c++ and piggyback c with message.
3. For receive event of message having logical value d, c := max(c,d) + 1.
`Note:C(e)<C(f) does not imply that e→f`
Consider, C(e) = 5 and C(f) = 7, we can only conclude that f did not happen before e. Two possibilities exist : e happened before or concurrent with f.

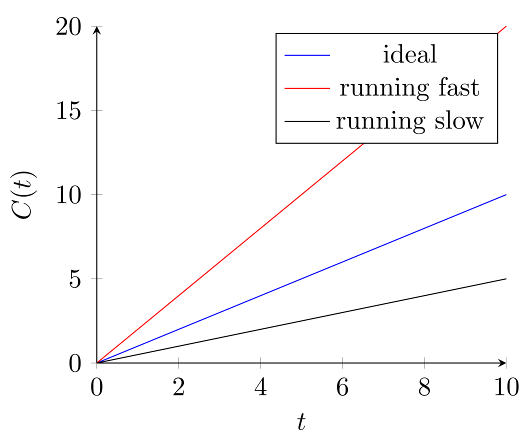
Sometimes, events need to have unique timestamps.In this case, (logical clock value, Process ID) becomes the timestamp. In case of a tie, event with smallest process id wins. Example



1.4 Physical Clock

1.4.1 Terminology

Let G = (V,E) be undirected graph representing topology.



Drift rate = $|1 - \frac{dc}{dt}| < \kappa$

τ = Synchronization Period (Every τ units of time, send your clock value to all your neighboring nodes)

1.4.2 Lamport's Physical Clock Algorithm

1. Assume, every sent message takes time in $[\mu, \mu + \xi]$ to reach destination.
2. On receiving a message timestamped with D, C:= max$(C, D+\mu)^+$ where + indicates that it is fractionally bigger (smallest level of granularity). Still, there exists uncertainty in synchronization(ϵ)
$|C_i(t) - C_j(t)| < \epsilon$
Diameter of the graph(d) = $max_{i,j}$ (length of the shortest path from i to j)

$\boxed{\epsilon \approx d(2\kappa\tau + \xi)}$

Advantage of physical clock is that is satisfies logic clock property.

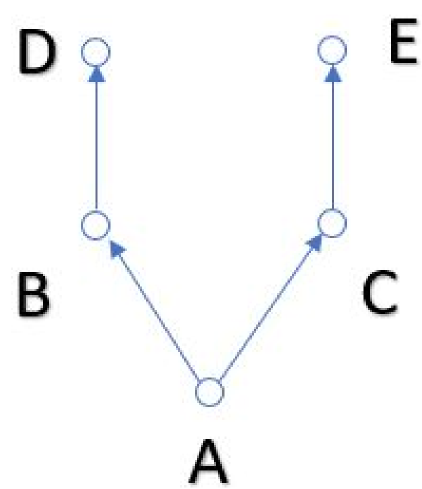
1.5 Down Sets

1.5.1 Terminology

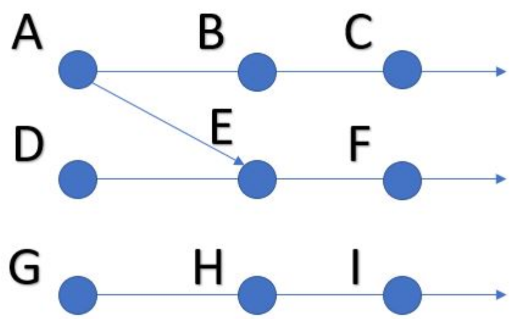
Let (X,$\leqslant$) be any poset. We call a subset Y $\subseteq$ X a down-set if:

$$(z \in Y) \wedge (y \leqslant z) \Rightarrow (y \in Y).$$

Example 1



Y = { D } is not a downset.
Y = { C, A } is a downset.
Example 2



D[E] = { A, D, E }

1.6 Principal Ideals

It refers to an ideal in a poset P generated by a single element x of P, which is to say the set of all elements less than or equal to x in P.

1.7 Vector clocks

When using vector clocks, the time domain is represented by a set of n-dimensional non-negative integer vector. A process Pi maintains a vector VT_i[1...n] where VT_i[i] is the local logical clock of Pi and describes the logical time progress at the process Pi.

Process Pi uses the following two rules R1 and R2 to update its clock:

- R1: Before executing an event, process Pi, updates its local logical time:

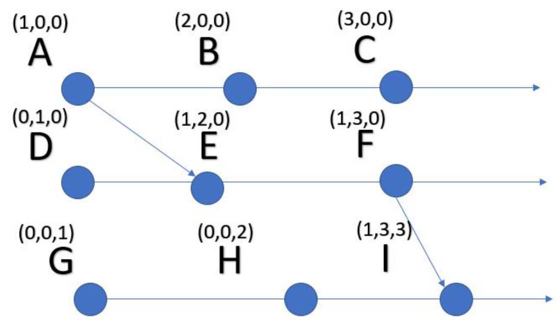
  $$VT_i[\mathrm{i}] = VT_i[\mathrm{i}\ ] + \mathrm{d} \text{ where } \mathrm{d} > 0$$

- R2: Each message m is piggybacked with the vector VT of the sender process at sending time. On the receipt of such a message (m,vt), process Pi executes the following actions:
  1. Update its global logical time as follows:

     $$1 \leq \mathrm{K} \leq \mathrm{n} : VT_i[\mathrm{i}] := \max(VT_i[\mathrm{k}]\,, \mathrm{VT}[\mathrm{k}]\,)$$

  2. Execute R1
  3. Deliver the message m

Example



1.8 Dilworth's Theorem(1950's)

1.8.1 Theorem

Any poset can be decomposed into w chains where w is the width of the poset. This is necessary and sufficient condition.

1.8.2 Proof

The basic idea is that when new element x comes up, we have to show that we can add it to one of the existing w chains.
`(To be continued...)`

References

[ACM78] Leslie, Lamport, Time, clocks, and the ordering of events in a distributed system, *Communications of the ACM* **21** (1978), pp. 558--565.

[Cambridge] Kshemkalyani, Ajay, Distributed Computing: Principals, algorithms, and systems (2008)