Time

Johan Montelius

KTH

HT15

Time

Why is time important?

The clock is not enough

In an asynchronous system clocks can note be completely trusted.

Nodes will not be completely synchronized.

We still need to:

- talk about before and after
- order events
- agree on order

Logical time

All events in one process are ordered.

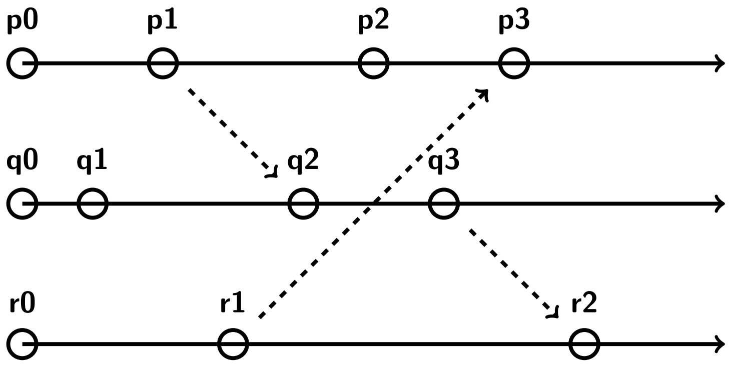
The sending of a message occurs before the receiving of the message.

Events in a distributed system are partially ordered.

The order is called “happened before”.

Logical time gives us a tool to talk about ordering without having to synchronize clocks.

Partial order



Lamport clock

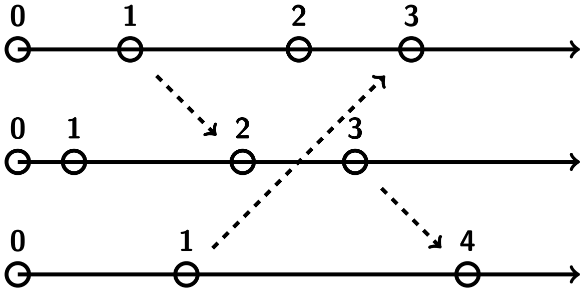
One counter per process:

- initially set to 0
- each process increment only its own clock
- sent messages are tagged with time stamp

Receiving a message:

- set the clock to the greatest of the internal clock and the time stamp of the message

Lamport clock



If *e1* **happened before** *e2* then the time stamp of *e1* is less than the time stamp of *e2*.

$$e_1 \textit{ happend-before } e_2 \rightarrow L(e_1) < L(e_2)$$

Let's play a game

Can we do better

We should be able to time stamp events so that we can capture the partial order.

We want to look at two time stamps and say:

if the time stamps are ordered then the events are ordered

$T(e_1) < T(e_2) \rightarrow e_1$ *happend-before* e_2

Vector clock

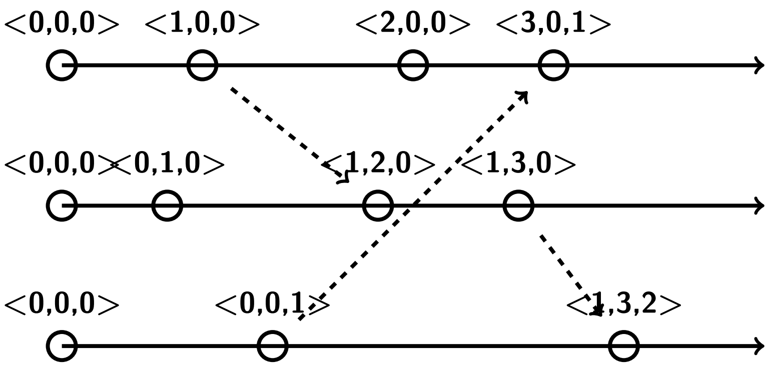
A vector with one counter per process:

- initially set to <0,....>
- each process increment only its own index
- sent messages are tagged with vector

Receiving a message:

- **merge** the internal clock and the time stamp of the message

Vector clock



$V(e_1) < V(e_2) \rightarrow e_1$ *happend-before* e_2

How do we define $<$ *over vector clocks?*

Pros and cons

The partial order is complete; we can look at the time stamp and determine if two events are ordered.

The vectors will take up a some space and could become a problem.

What should we do if more processes come and leave, there is no easy mechanism to add new clocks to the system.

Vector clocks could be over-kill.

Alternative implementation

Summary

If we can not trust real clocks to be synchronized we have to use something else.

Logical time captures what we need:

- Lamport clock: sound
- Vector clock: complete

Implementation issues:

- do we have to time stamp everything
- how do we handle new processes