Vector Clocks

Problem: Detecting causal relations

If L(e) < L(e’)

– Cannot conclude that e→e’

Looking at Lamport timestamps

– Cannot conclude which events are causally related

Solution: use a **vector clock**

Vector Clocks

- Developed independently by Fidge, Mattern and Schmuck.

- Time is represented by a set of n-dimensional non-negative integer vectors.

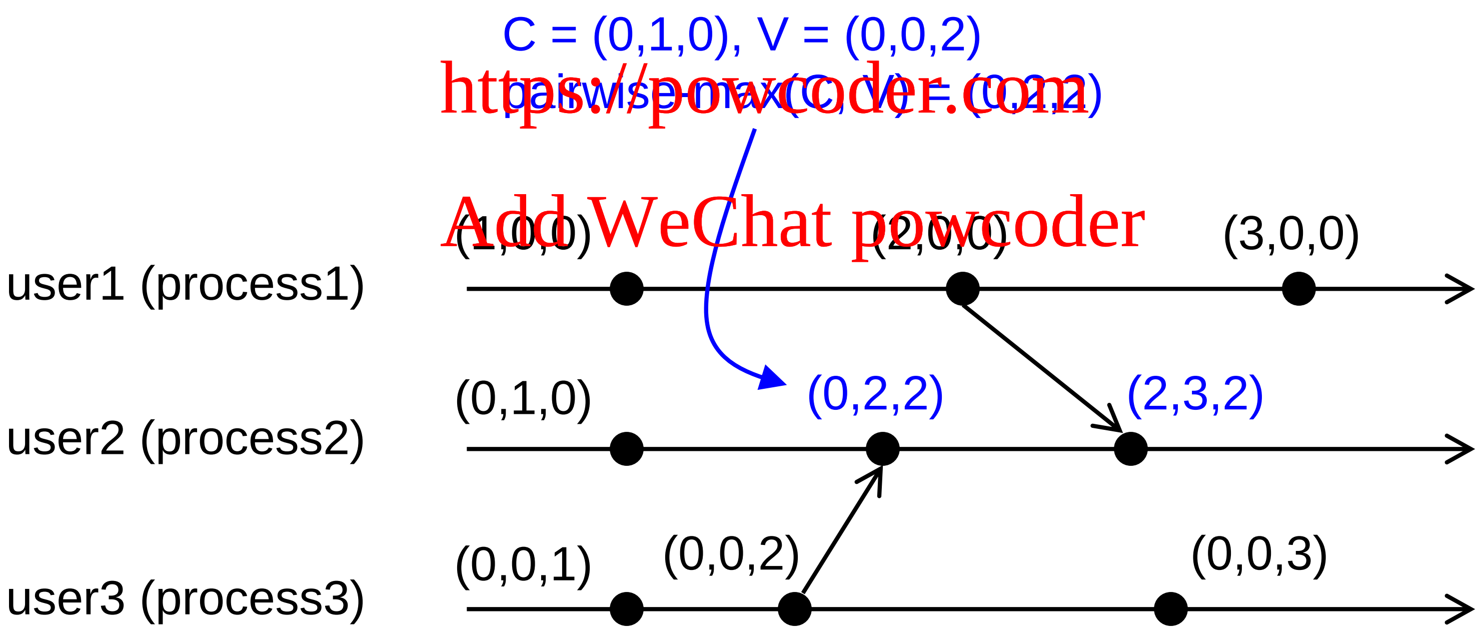
- Each process has a clock C_i consisting of a vector of length *n*, the total number of processes vt[1..n]. vt[j] is the local logical clock of Pj and describes the logical time progress at process Pj

Vector Clock Protocol

- P_i ticks by incrementing its own component of its clock
  - $C_i[i]$ += 1
- Timestamp C(e) of event e is the clock value after ticking
- Each message is piggybacked with the local vector u
- Recipient updates local vector to max of u and local vector v

Vector Clock Protocol

- Each process i has a local vector C
- Increment C[i] at each “local computation” and “send” event
- When sending a message, vector clock value V is attached to the message. At each “receive” event, C = pairwise-max(C, V)



C = (0,1,0), V = (0,0,2)
pairwise-max(C, V) = (0,2,2)

Vector clocks Protocol

1. Vector initialized to 0 at each process

   $V_i\,[j] = 0$ for $i, j = 1, \ldots, N$

2. Process increments its element of the vector in local vector before event:

   $V_i\,[i] = V_i\,[i] + 1$

3. Piggyback V_i with every message sent from process P_i

4. When P_j receives message, compares vectors element by element and sets local vector to higher of two values

   $V_j\,[k] = \max(V_j\,[k], V_i\,[k])$ for k=1, …, N

Comparing vector timestamps

Define

$V = V'$ iff $V[i] = V'[i]$ for $i = 1 \ldots N$

$V \leq V'$ iff $V[i] \leq V'[i]$ for $i = 1 \ldots N$

$V < V'$ iff $V \leq V'$ and $V \neq V'$

For any two events e, e'

e → e' if and only if $V(e) < V(e')$

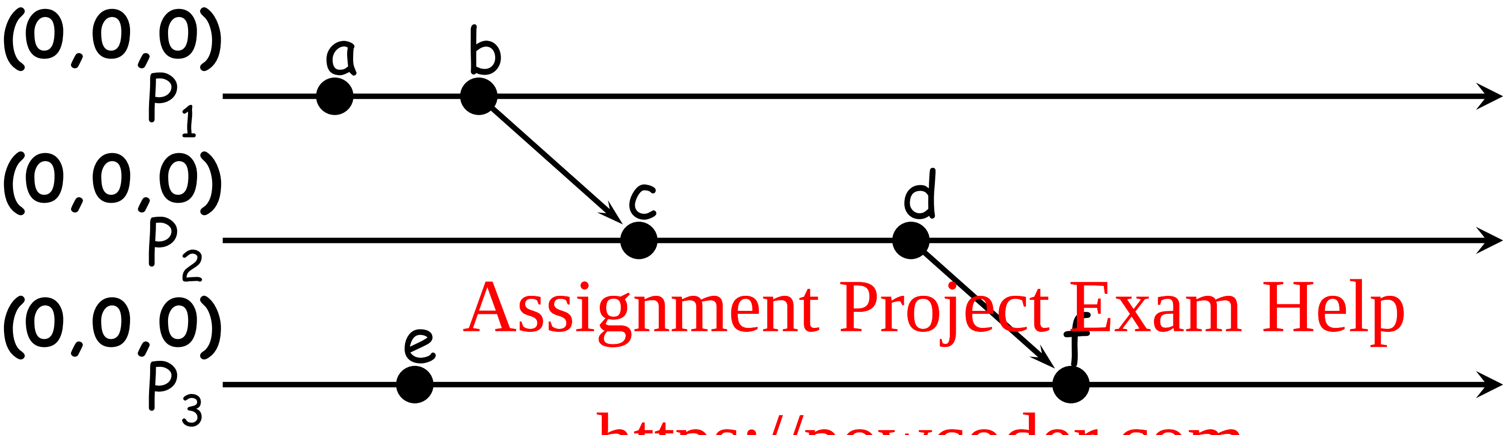
Two events are **concurrent** if **neither**

$V(e) \leq V(e')$ nor $V(e') \leq V(e)$

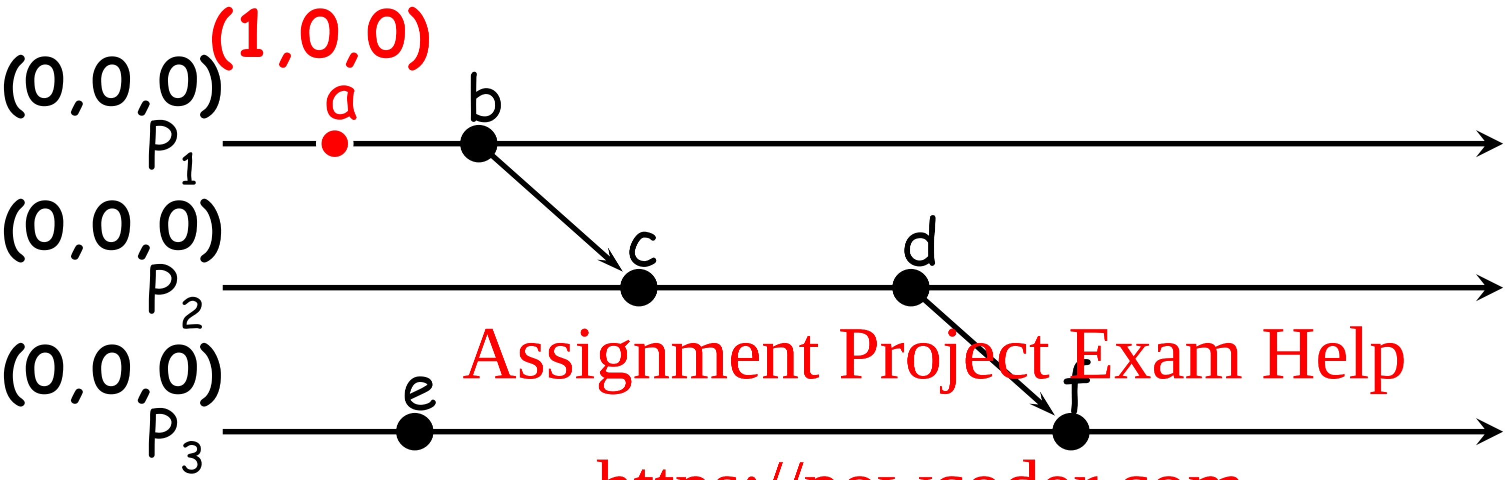
Structure of the Vector Clock

- In order to determine if two events e,e’ are causally related or not, just take their vector timestamps V(e) and V(e’):
  - if V(e)<V(e’) ∨ V(e’)<V(e), then e’ and e *are causally related*
  - Otherwise, they *are concurrent.*

Vector timestamps

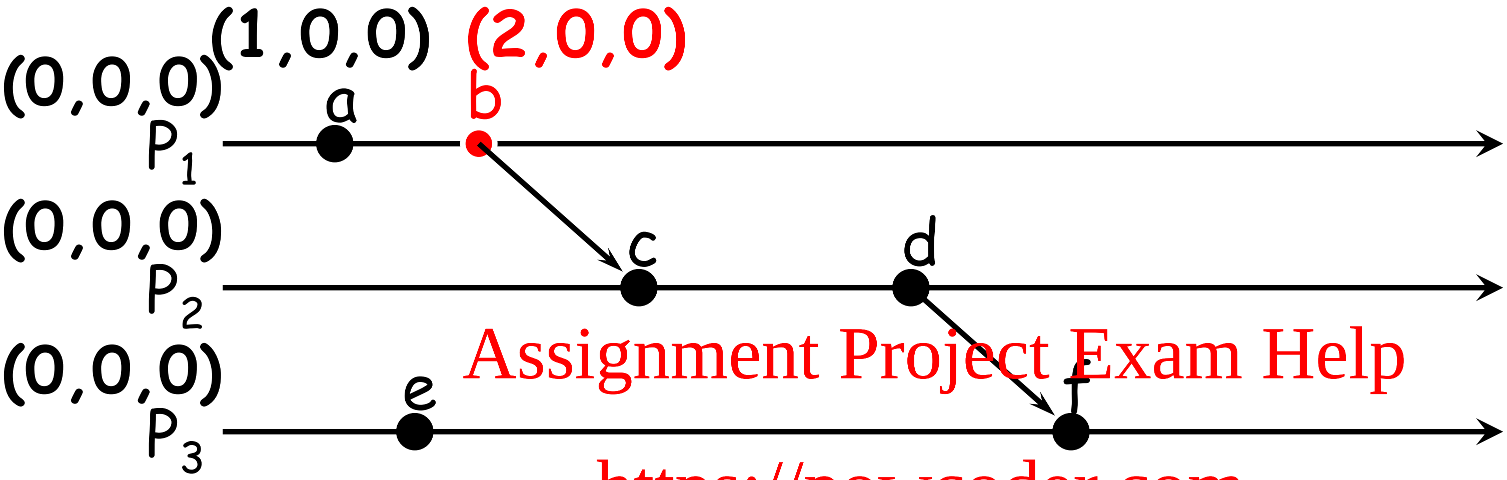


Vector timestamps



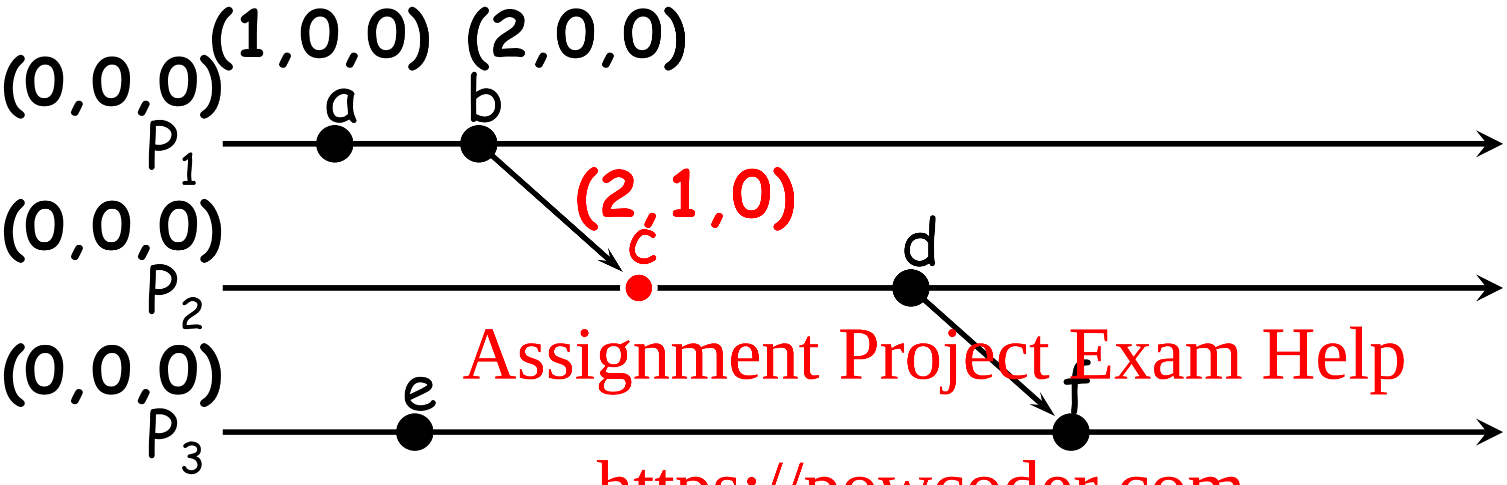
Event	timestamp
a	(1,0,0)

Vector timestamps



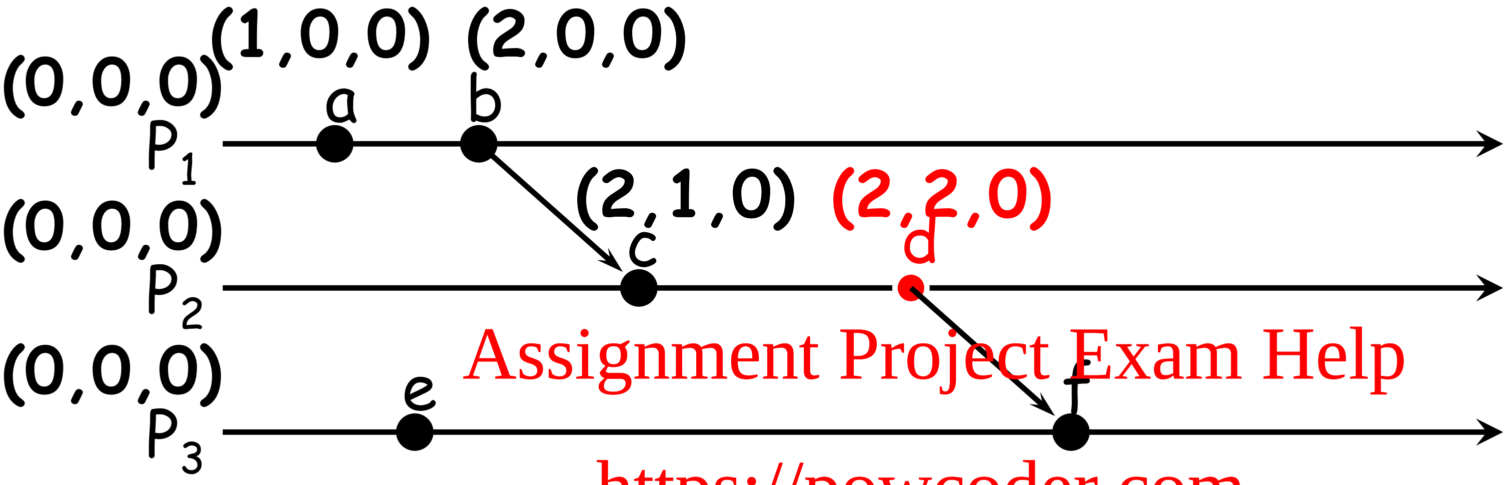
Event	timestamp
a	(1,0,0)
b	(2,0,0)

Vector timestamps



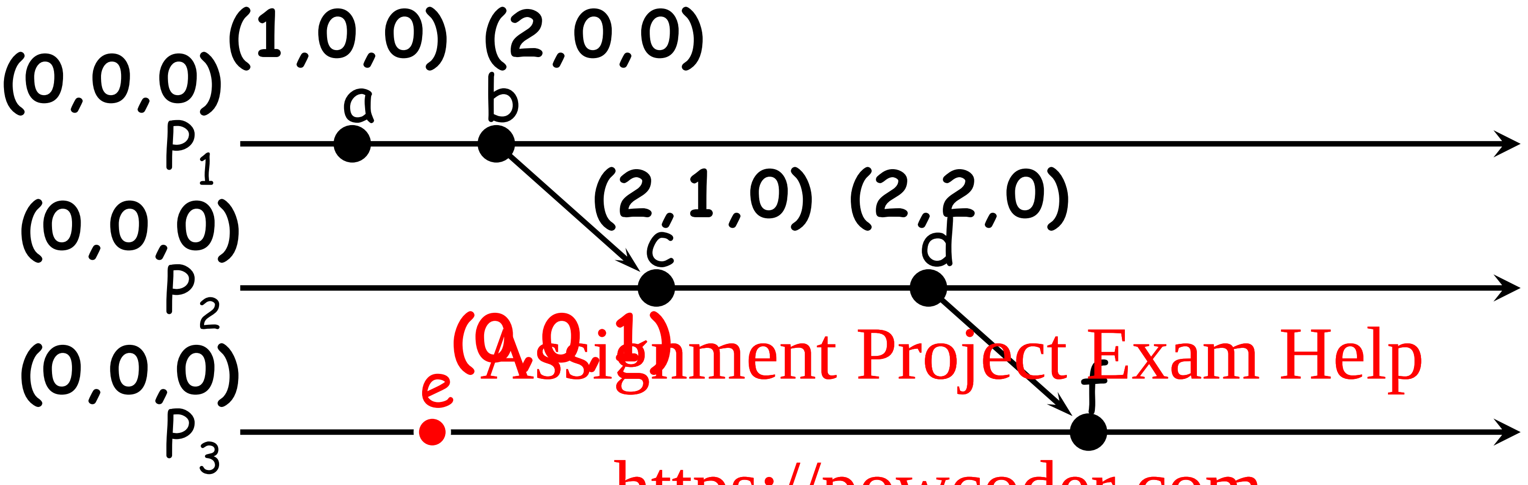
Event	timestamp
a	(1,0,0)
b	(2,0,0)
c	(2,1,0)

Vector timestamps



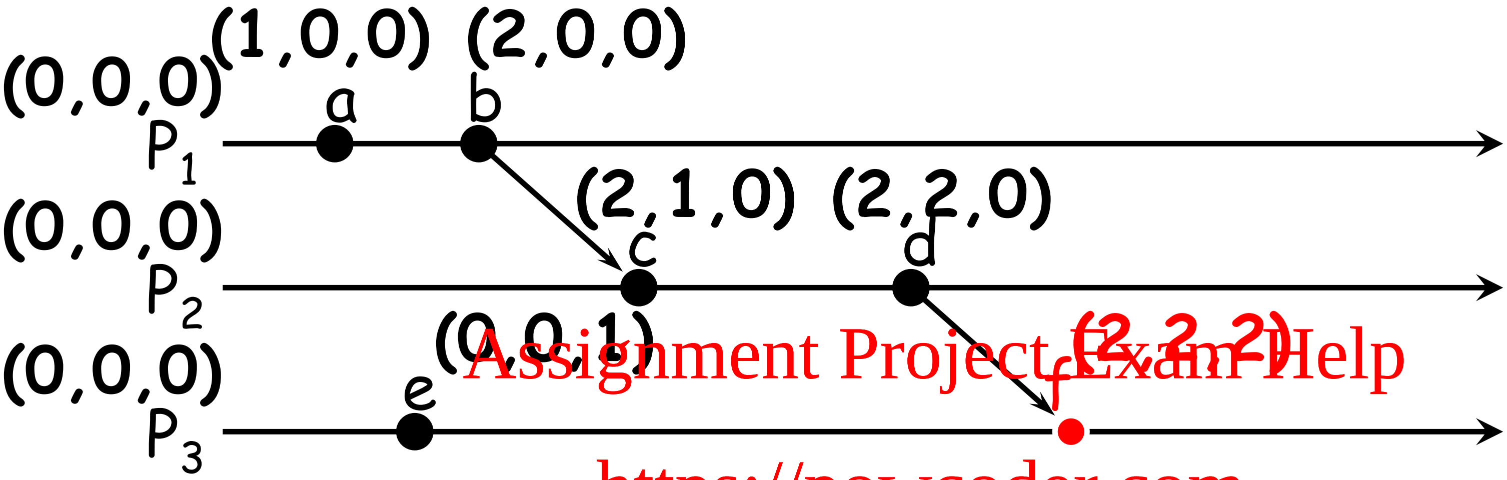
Event	timestamp
a	(1,0,0)
b	(2,0,0)
c	(2,1,0)
d	(2,2,0)

Vector timestamps



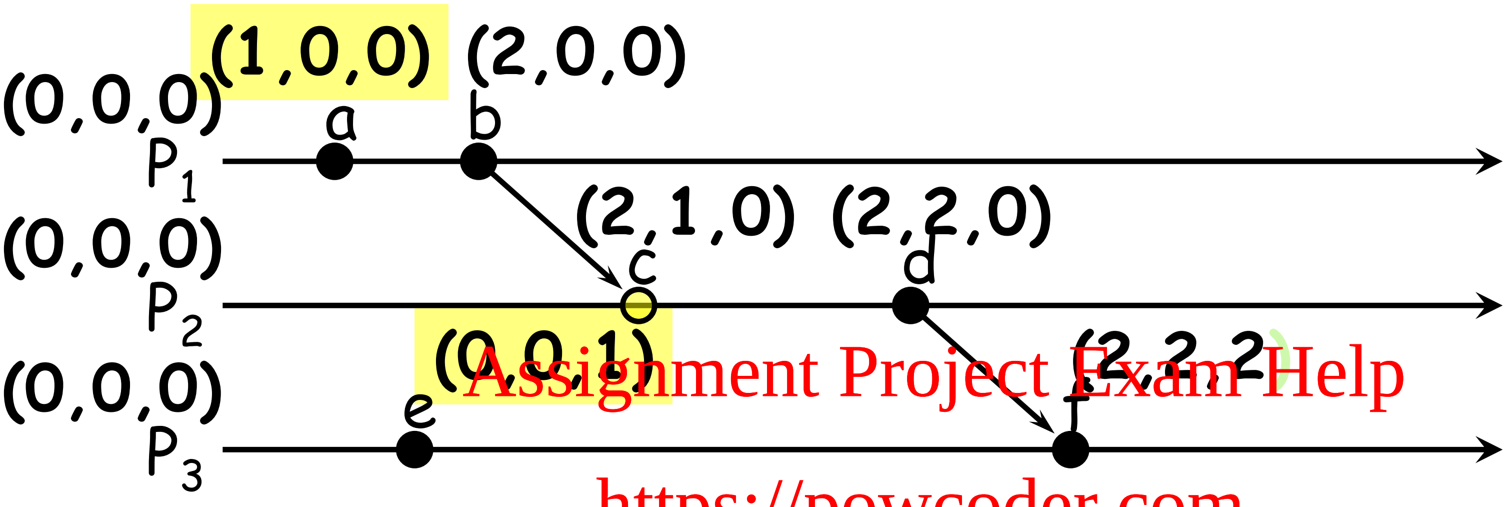
Event	timestamp
a	(1,0,0)
b	(2,0,0)
c	(2,1,0)
d	(2,2,0)
e	(0,0,1)

Vector timestamps



Event	timestamp
a	(1,0,0)
b	(2,0,0)
c	(2,1,0)
d	(2,2,0)
e	(0,0,1)
f	(2,2,2)

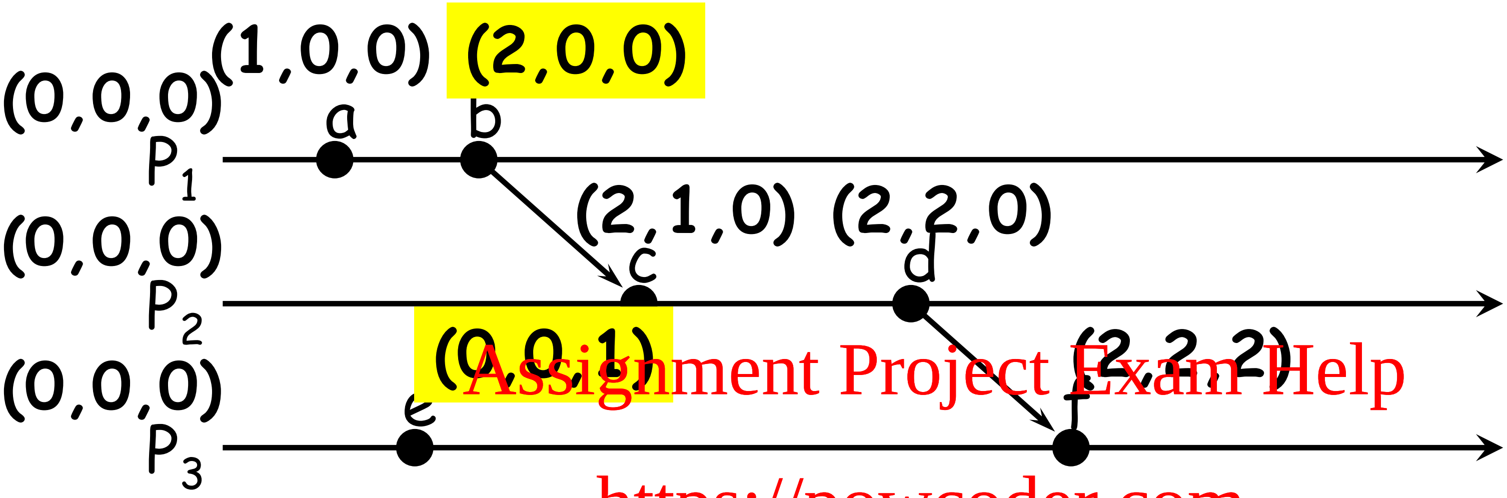
Vector timestamps



Event	timestamp
a	(1,0,0)
b	(2,0,0)
c	(2,1,0)
d	(2,2,0)
e	(0,0,1)
f	(2,2,2)

concurrent events

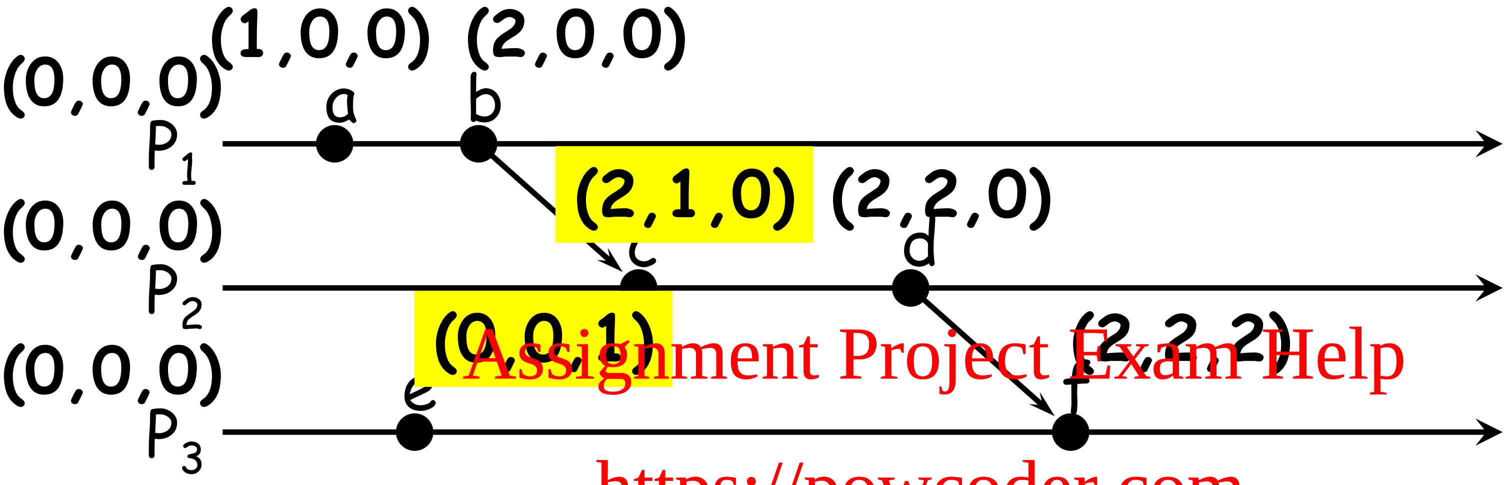
Vector timestamps



Event	timestamp
a	(1,0,0)
b	(2,0,0)
c	(2,1,0)
d	(2,2,0)
e	(0,0,1)
f	(2,2,2)

concurrent events

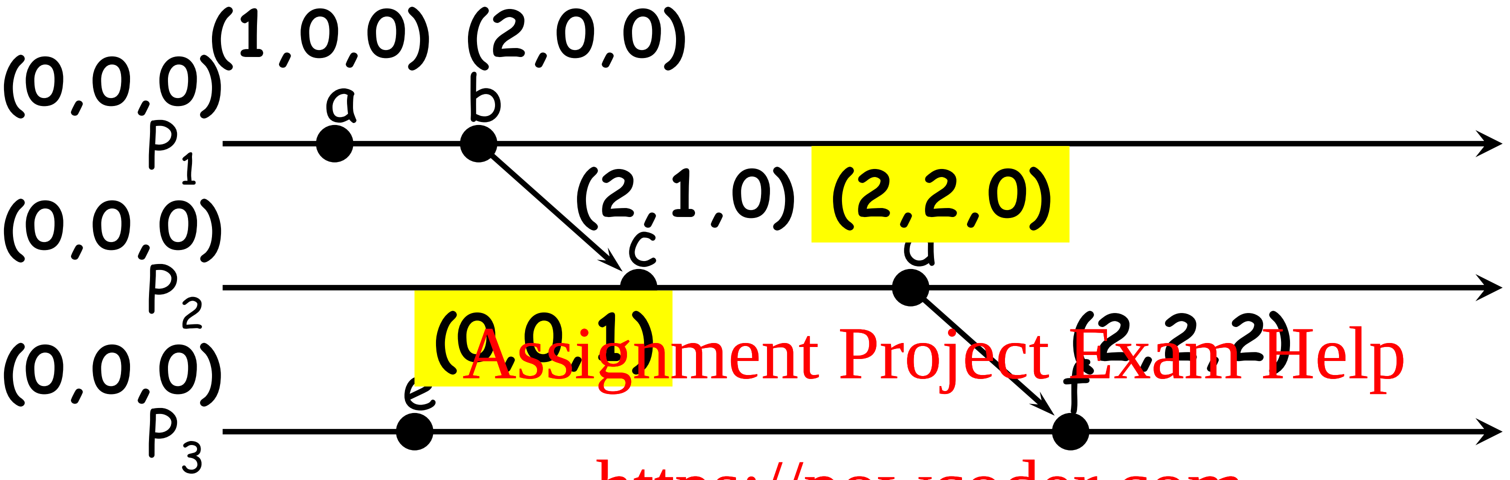
Vector timestamps



Event	timestamp
a	(1,0,0)
b	(2,0,0)
c	(2,1,0)
d	(2,2,0)
e	(0,0,1)
f	(2,2,2)

concurrent events

Vector timestamps



Event	timestamp
a	(1,0,0)
b	(2,0,0)
c	(2,1,0)
d	(2,2,0)
e	(0,0,1)
f	(2,2,2)

concurrent events

Vector Clocks

- Because of transitive nature, a process *may receive* time updates about clocks in non-neighboring process
- Since process P_i can advance the i^{th} component of local vector, it always has most accurate knowledge for local events
  - At any instant: $\forall i,j$: $C_i[i] \geq C_j[i]$

Strong Consistency

- For any two events e, e’

  $e \rightarrow e'$ if and only if $V(e) < V(e')$

- The system of vector clocks is strongly consistent; thus, by examining the vector timestamp of two events, we can determine if the events are causally related.

- Could we have used integers?
  - NO: Charron-Bost showed that for this property to hold, the dimension of vector clocks cannot be less than n, the total number of processes in the computation.