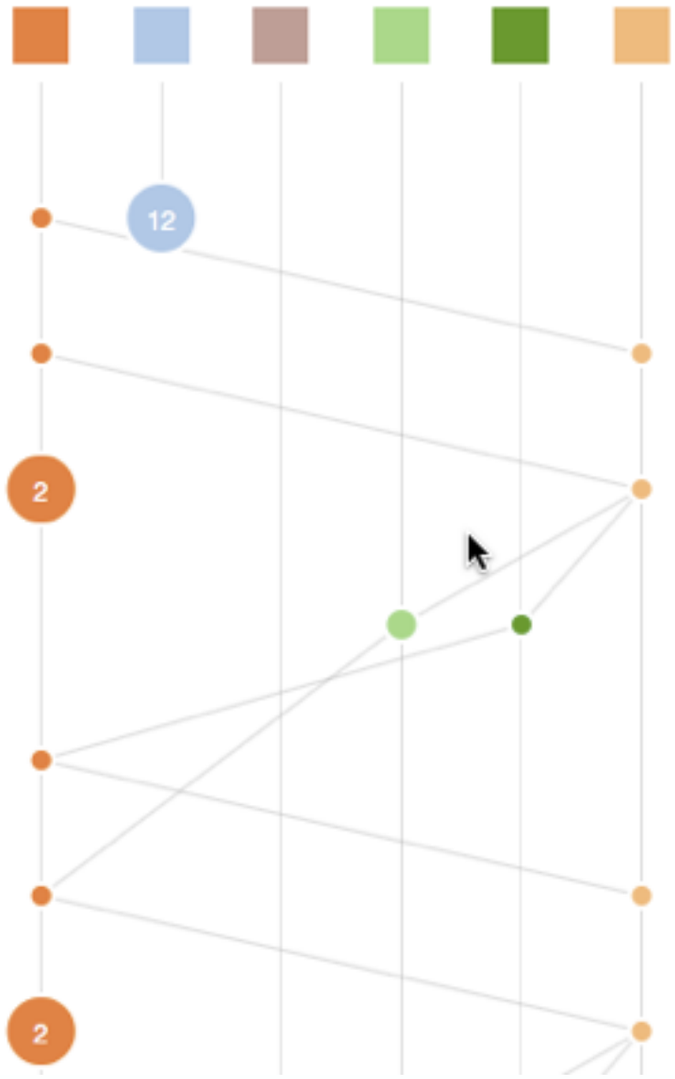



Brief lecture and
in-class research study

# A tool to visualize logs of distributed systems

University of British Columbia

UMass Amherst

University of Washington

# Distributed log analysis: hard

src : 2, dst : 0, timestamp : 0, type : prepare
src : 2, dst : 1, timestamp : 1, type : prepare
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
src : 0, dst : 2, timestamp : 6, type : ack
src : 1, dst : 2, timestamp : 7, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 2, dst : 0, timestamp : 0, type : prepare
src : 2, dst : 1, timestamp : 1, type : prepare
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
src : 0, dst : 2, timestamp : 6, type : ack
src : 1, dst : 2, timestamp : 7, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare

log1.txt

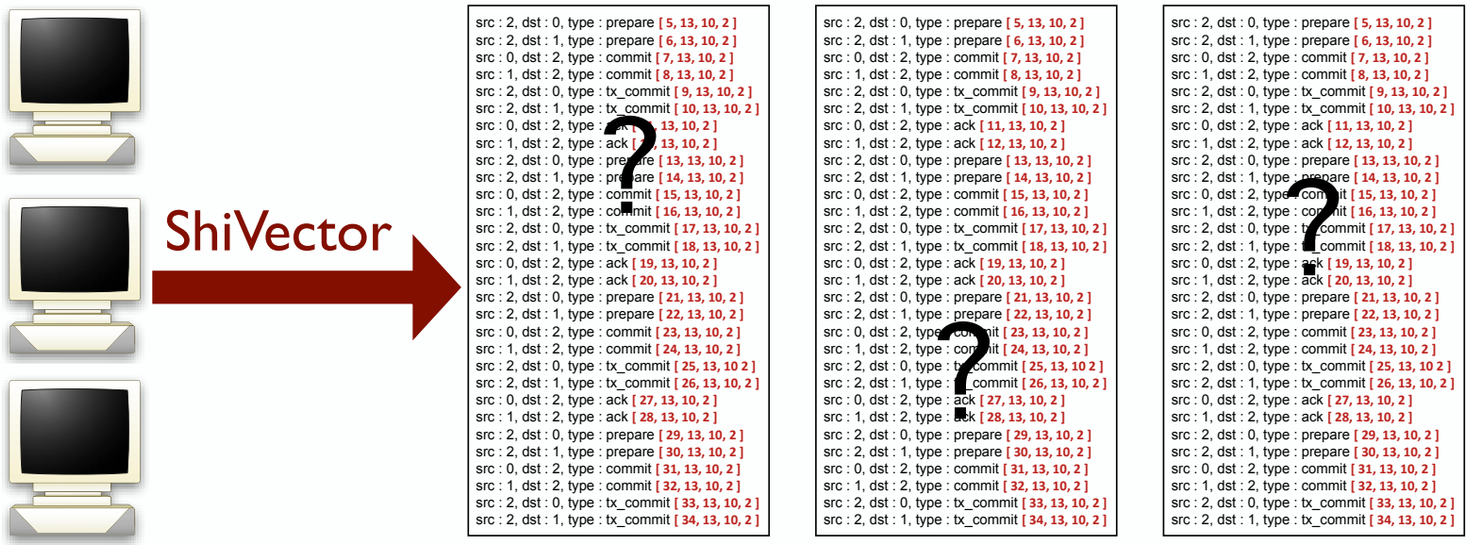
src : 2, dst : 0, timestamp : 0, type : prepare
src : 2, dst : 1, timestamp : 1, type : prepare
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
src : 0, dst : 2, timestamp : 6, type : ack
src : 1, dst : 2, timestamp : 7, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 2, dst : 0, timestamp : 0, type : prepare
src : 2, dst : 1, timestamp : 1, type : prepare
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
src : 0, dst : 2, timestamp : 6, type : ack
src : 1, dst : 2, timestamp : 7, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare

log2.txt

src : 2, dst : 0, timestamp : 0, type : prepare
src : 2, dst : 1, timestamp : 1, type : prepare
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
src : 0, dst : 2, timestamp : 6, type : ack
src : 1, dst : 2, timestamp : 7, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 2, dst : 0, timestamp : 0, type : prepare
src : 2, dst : 1, timestamp : 1, type : prepare
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
src : 0, dst : 2, timestamp : 6, type : ack
src : 1, dst : 2, timestamp : 7, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare

log3.txt

# Instrument the system to record time

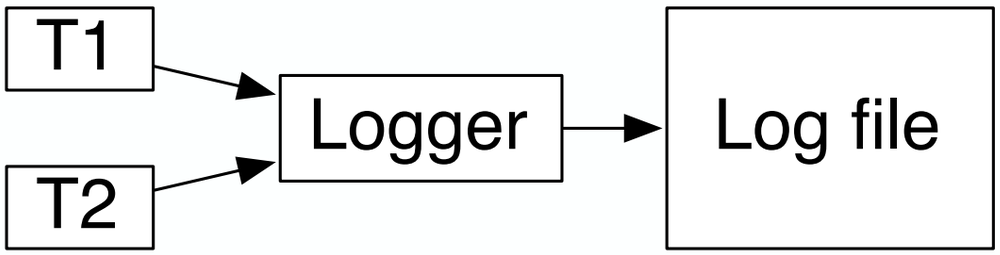



# Time as a partial order

# Concurrency and logging

- A system with two threads: T1, T2
  - T1 generates event (a), T2 generates event (b)

- Logging pipeline:



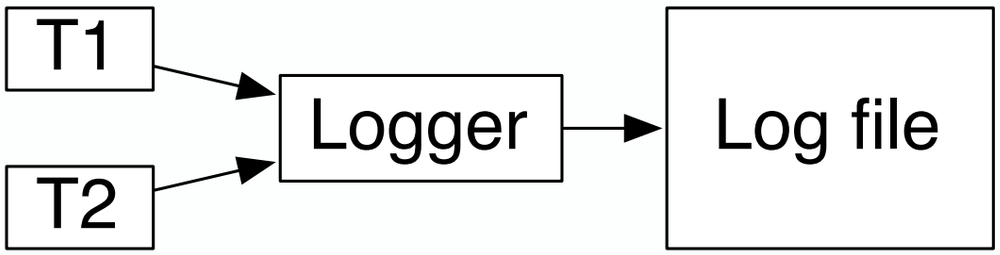

- Generated log file:

| | |
|---|---|
| 1 | a |
| 2 | b |

# Concurrency and logging

- A system with two threads: T1, T2
  - T1 generates event (a), T2 generates event (b)

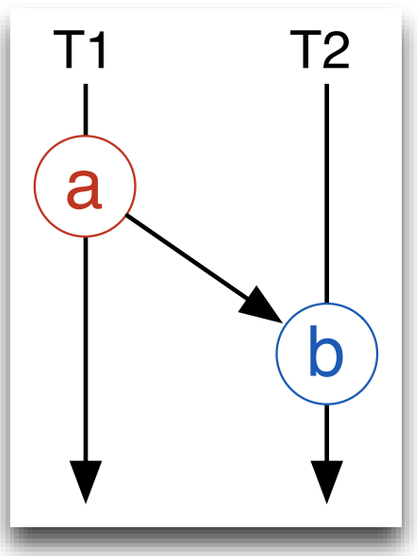
- Logging pipeline:

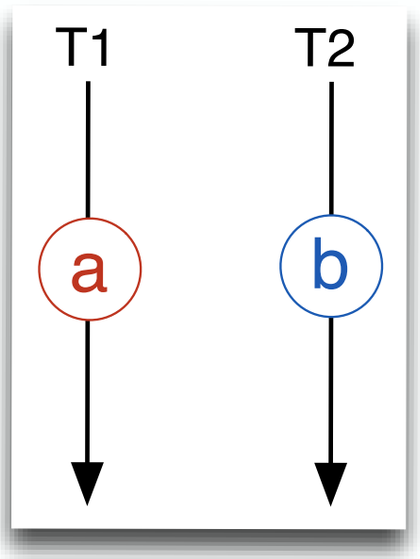


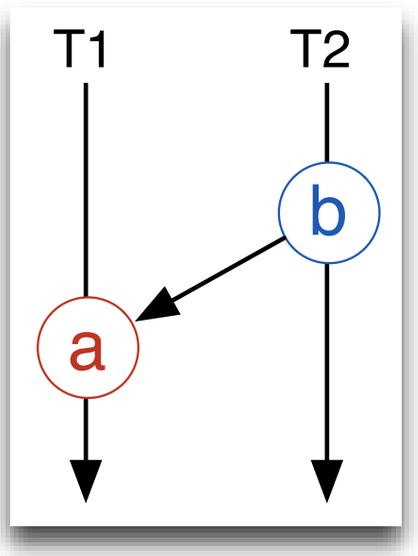

- Generated log file:

| | |
|---|---|
| 1 | a |
| 2 | b |

Which of these three systems generated the log?



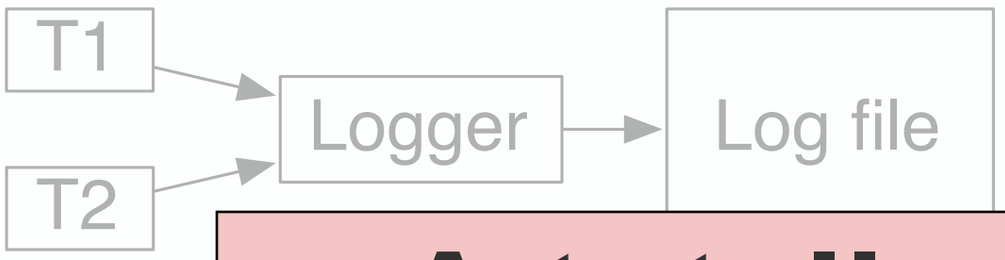



# Concurrency and logging

- A system with two threads: T1, T2
  - T1 generates event (a), T2 generates event (b)

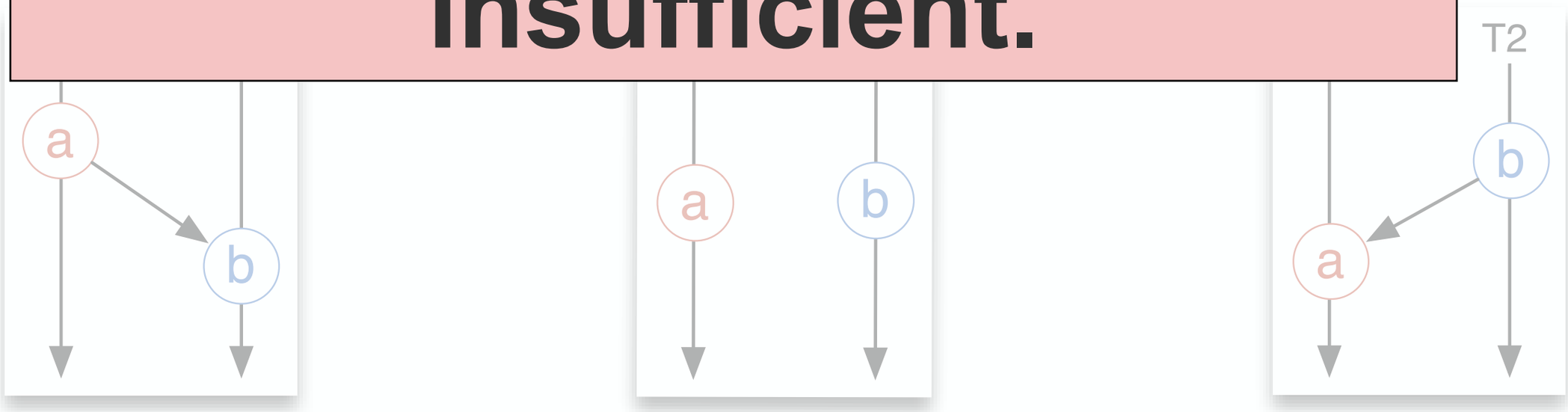
- Logging pipeline:




- Generated log file:

| | |
|---|---|
| 1 | a |
| 2 | b |

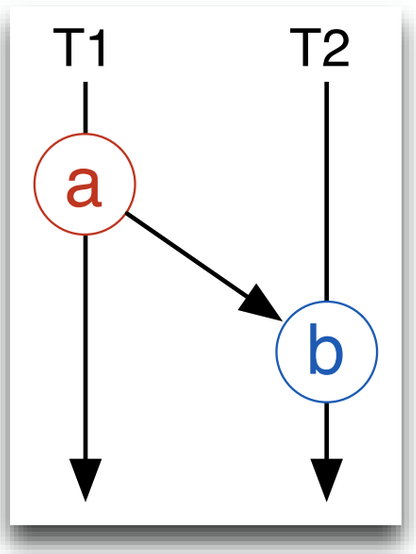
**A totally ordered log is insufficient.**



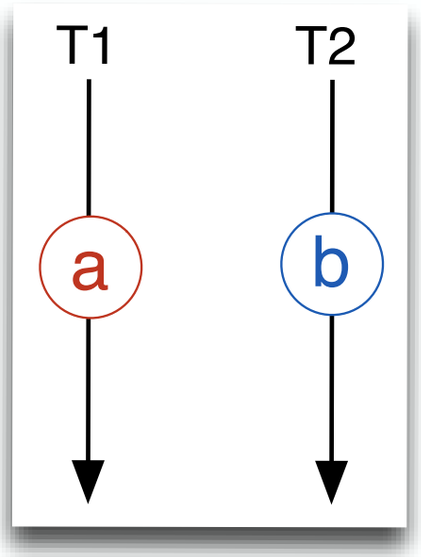
# Logging the partial order

- We know how to do this
  - Lamport defined the happens-before relation in 1978
  - Operationalized with vector clocks in 1988, 1989

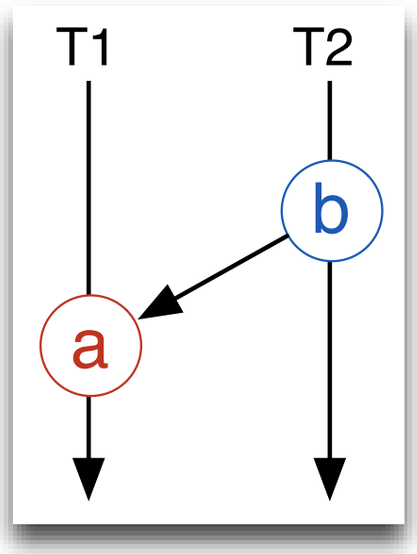
| | |
|---|---|
| [1,0] | a |
| [1,1] | b |




| | |
|---|---|
| [1,0] | a |
| [0,1] | b |

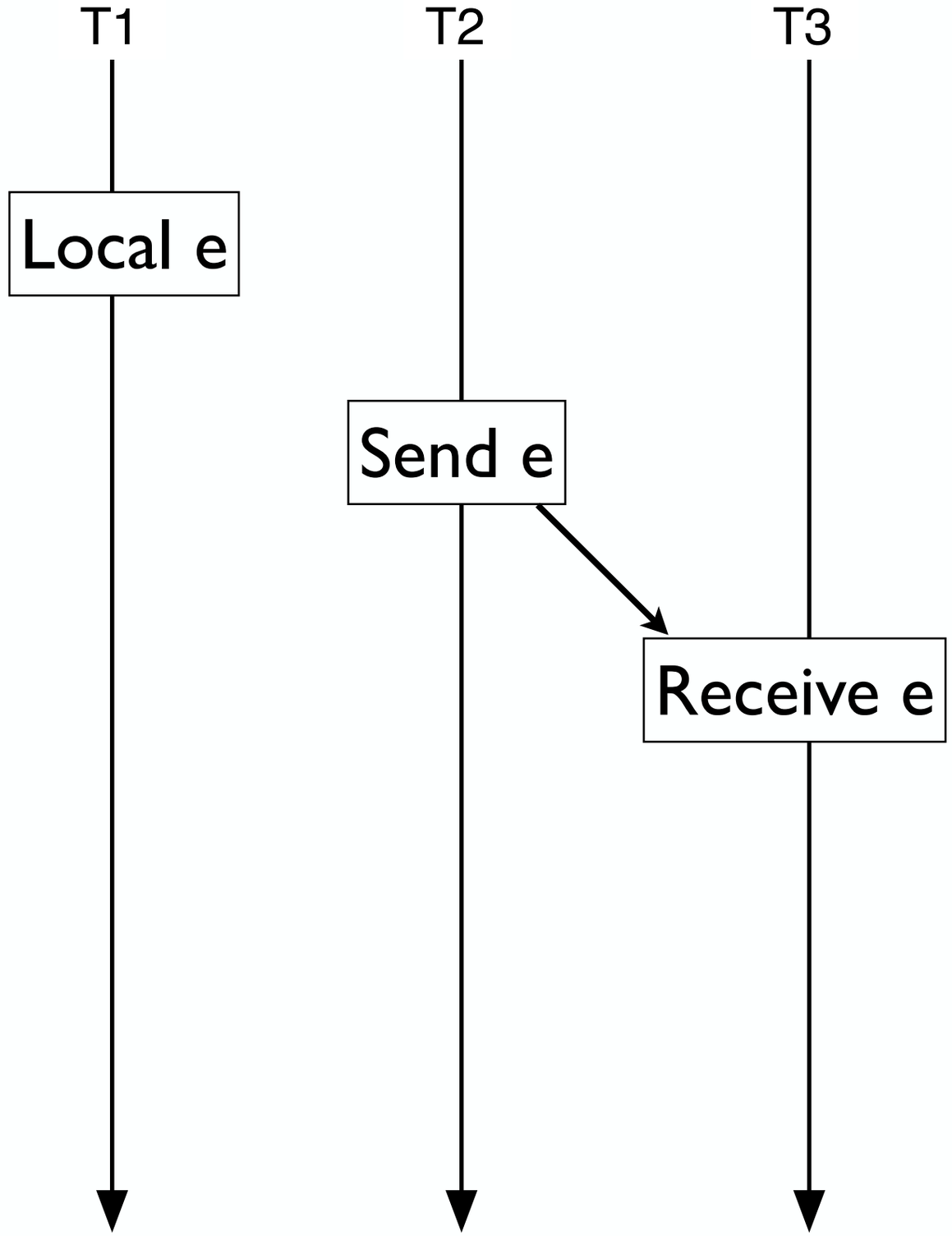



| | |
|---|---|
| [1,1] | a |
| [0,1] | b |



# Vector clocks overview

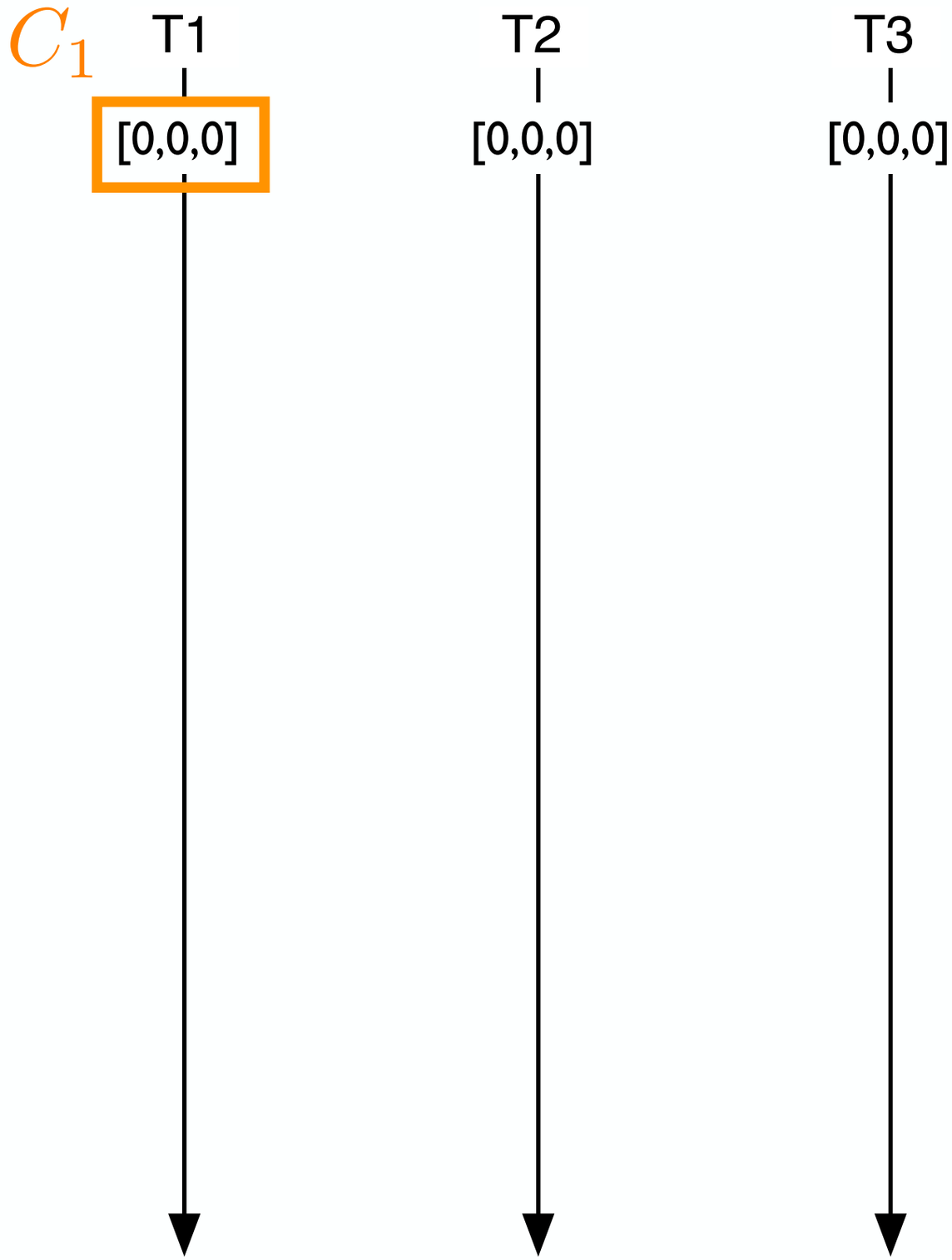
- Three threads as an example
- Three kinds of events
  - Local
  - Message send
  - Message receive




| Fidge, ACSS 1988 | Mattern, PDA 1989 |
|---|---|

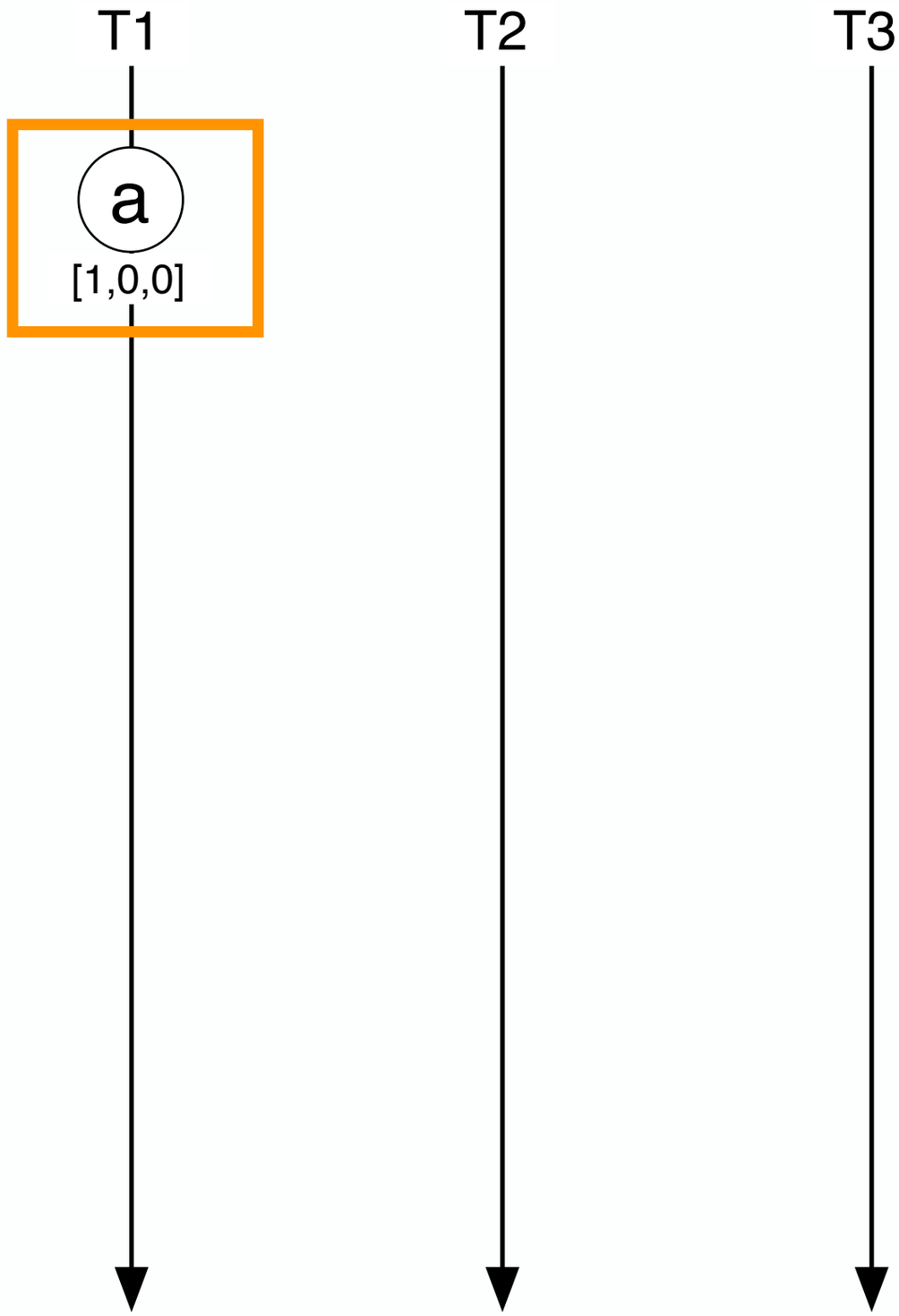
# Vector clocks overview

- Thread i vector clock is $C_i$
- Each thread initializes its vector clock to [0,0,0]



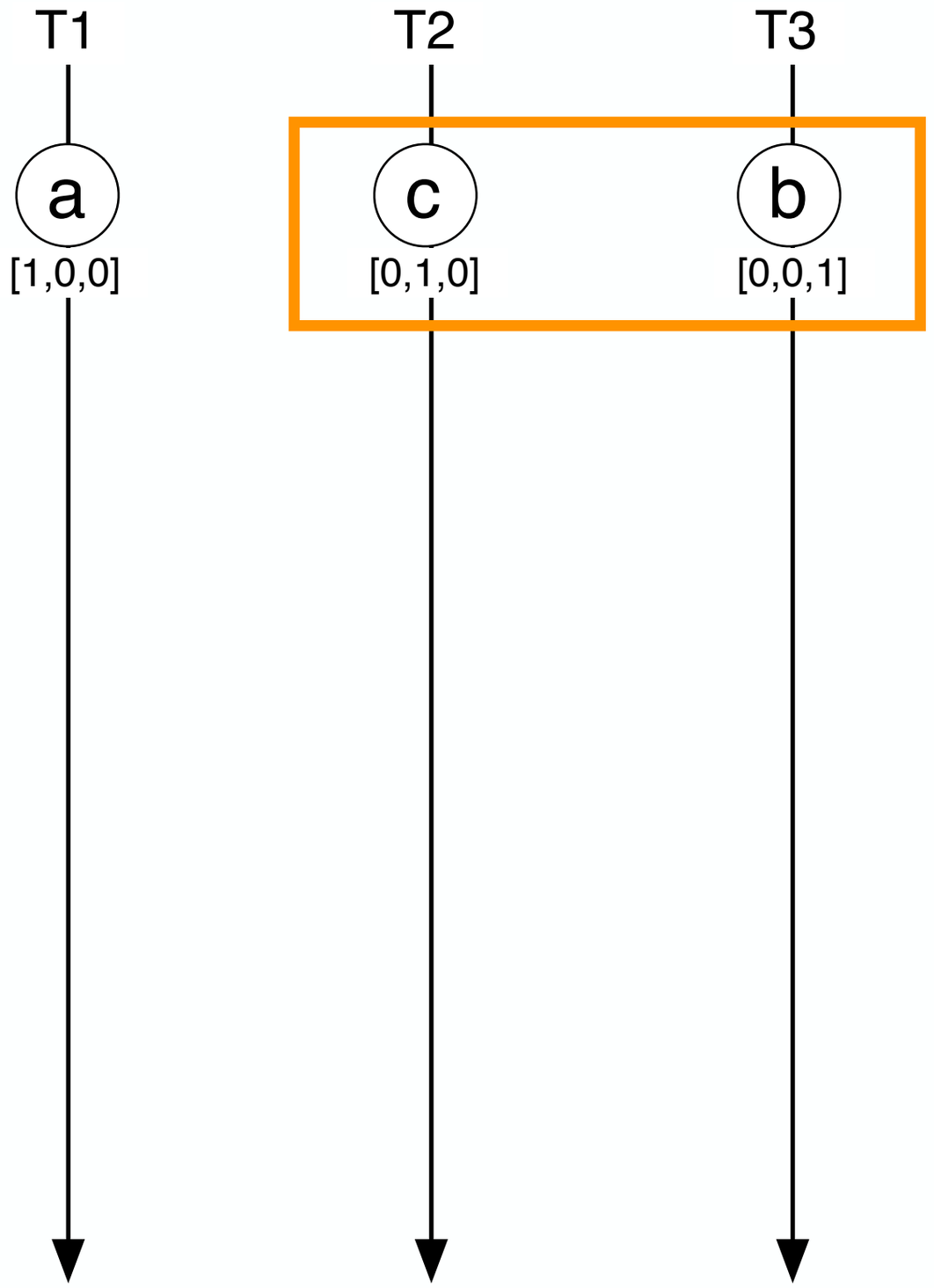
# Vector clocks overview

- Thread i vector clock is $C_i$
- Each thread initializes its vector clock to [0,0,0]
- On a local event at thread i the thread increments its local clock index $C_i[i] + +$



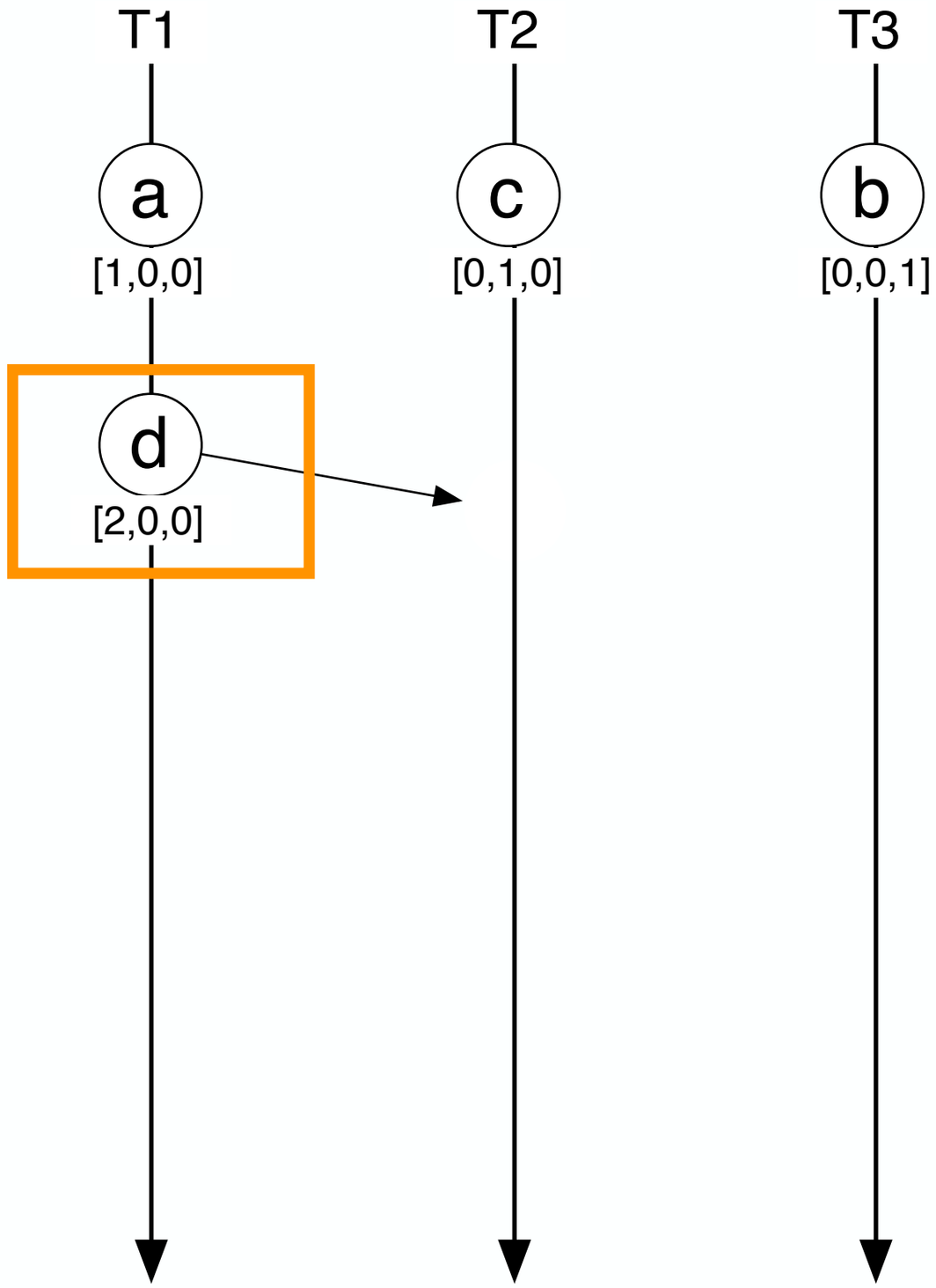
# Vector clocks overview

- Thread i vector clock is $C_i$
- Each thread initializes its vector clock to [0,0,0]
- On a local event at thread i the thread increments its local clock index $C_i[i] + +$



# Vector clocks overview

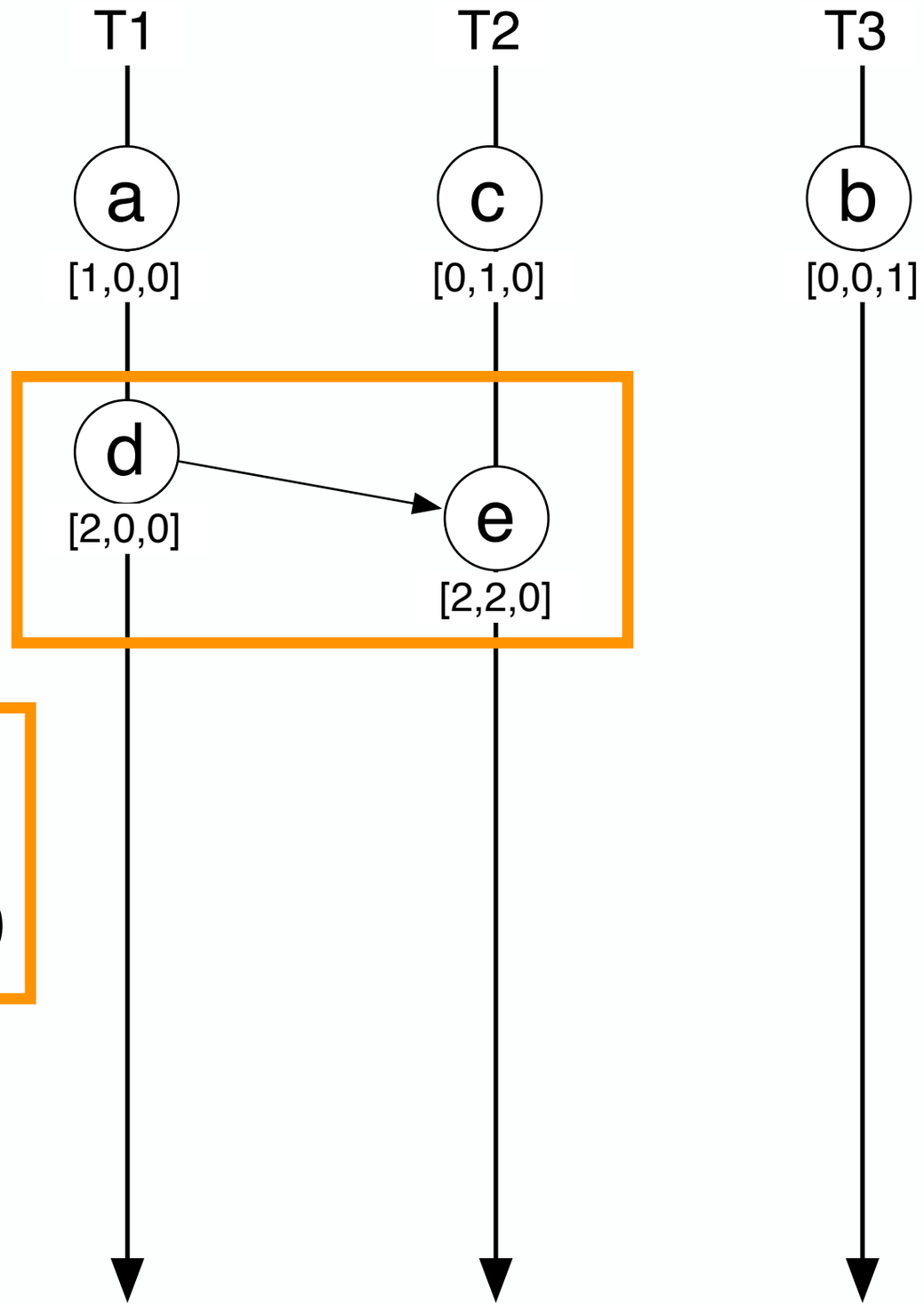
- Thread i vector clock is $C_i$
- Each thread initializes its vector clock to [0,0,0]
- On a local event at thread i the thread increments its local clock index $C_i[i]++$



# Vector clocks overview

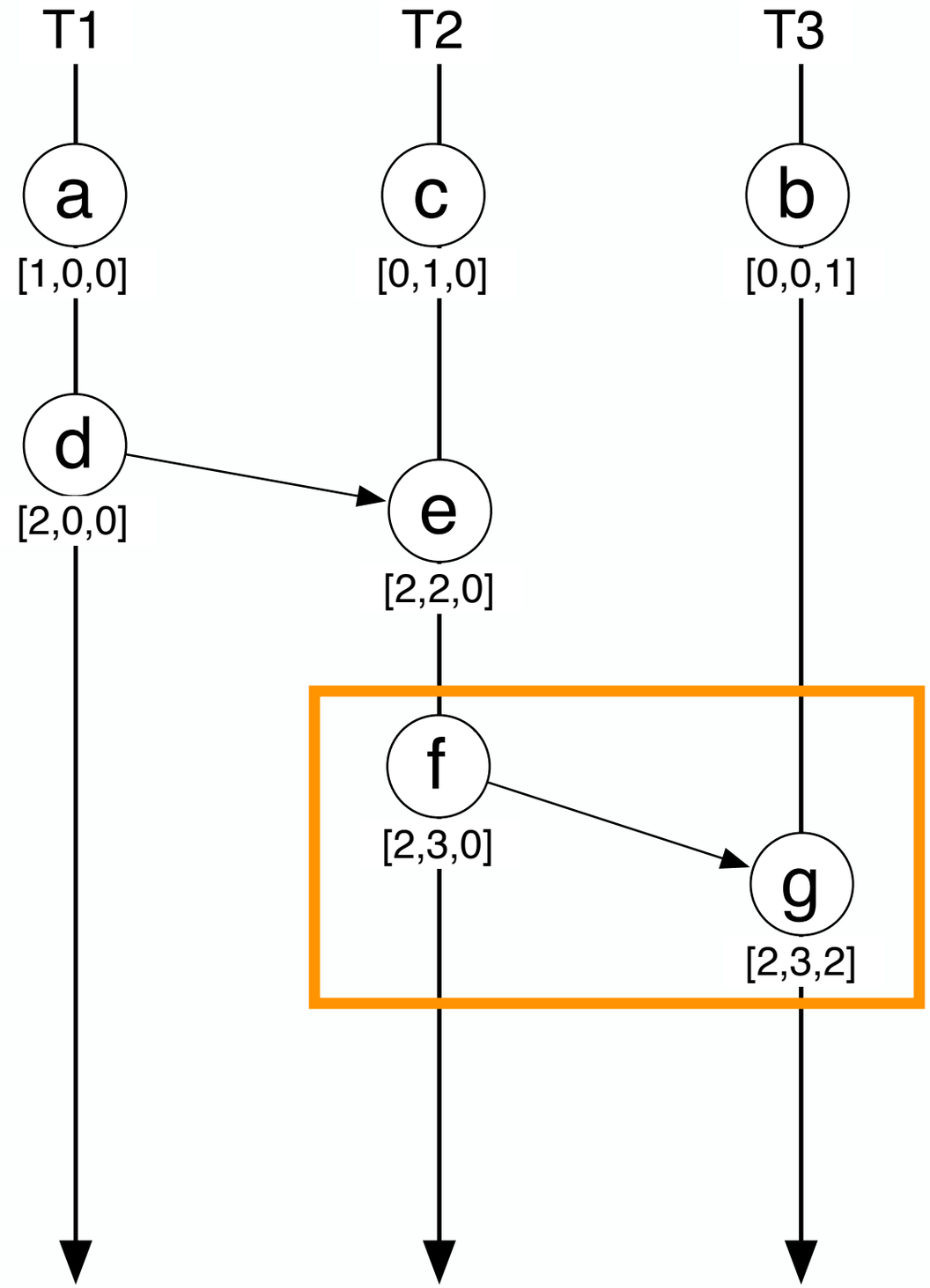
- Thread i vector clock is $C_i$
- Each thread initializes its vector clock to [0,0,0]
- On a local event at thread i the thread increments its local clock index $C_i[i] + +$

- Sender i transmits its vector to recipient j, which updates its vector with $C_j[h] = \max(C_j[h], C_i[h])$



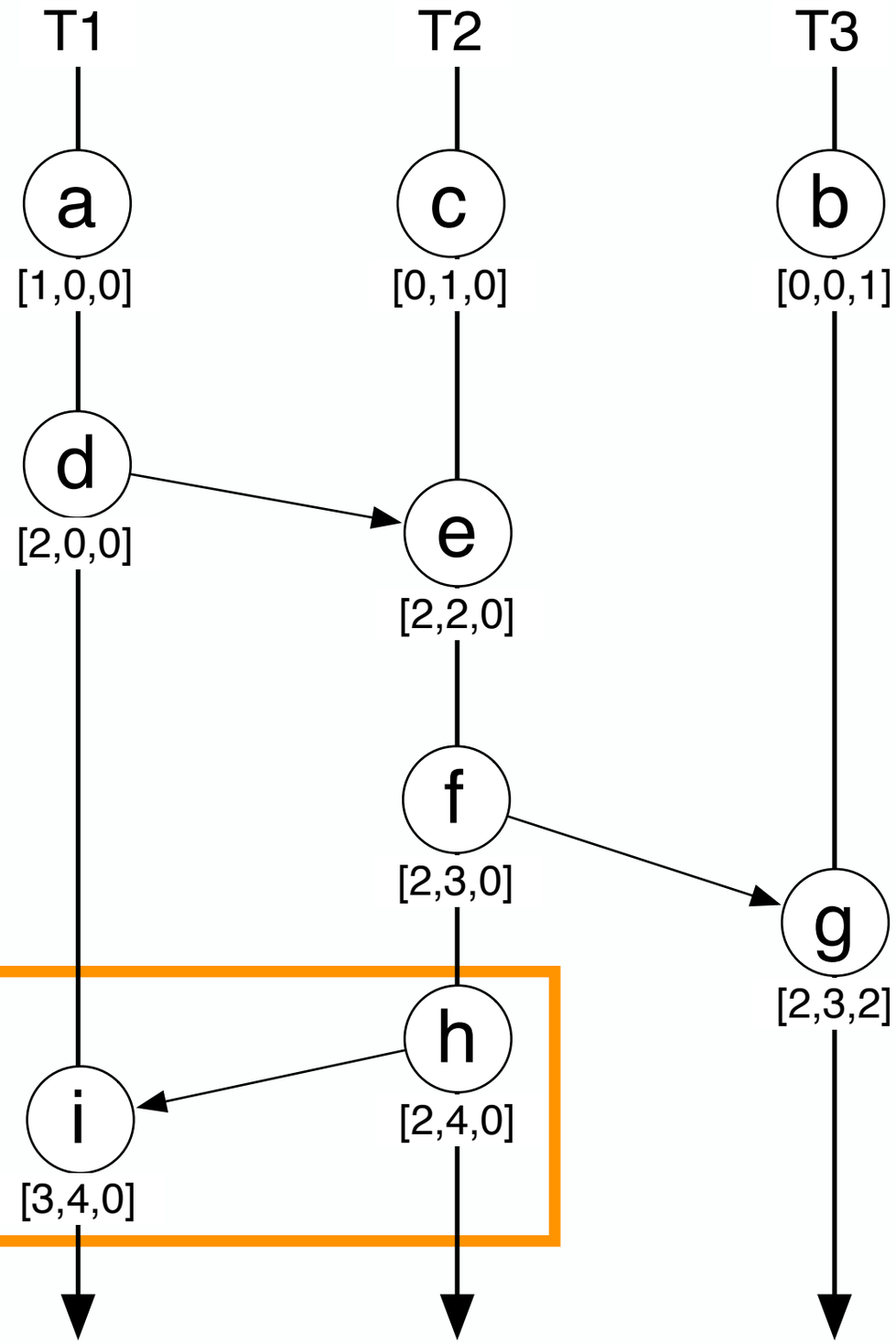
# Vector clocks overview

- Thread i vector clock is $C_i$
- Each thread initializes its vector clock to [0,0,0]
- On a local event at thread i the thread increments its local clock index $C_i[i] + +$
- Sender i transmits its vector to recipient j, which updates its vector with $C_j[h] = \max(C_j[h], C_i[h])$



# Vector clocks overview

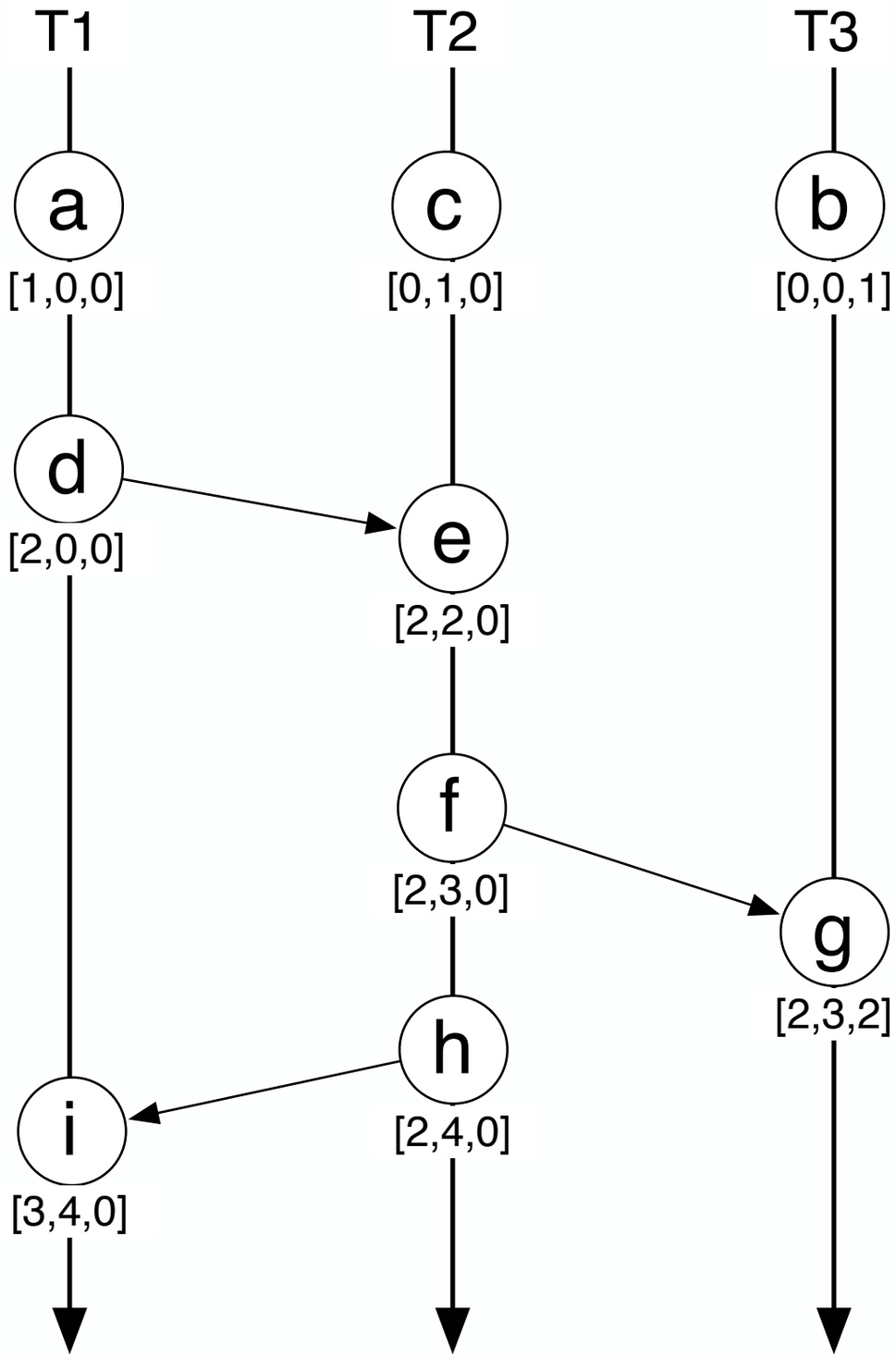
- Thread i vector clock is $C_i$
- Each thread initializes its vector clock to [0,0,0]
- On a local event at thread i the thread increments its local clock index $C_i[i] + +$
- Sender i transmits its vector to recipient j, which updates its vector with $C_j[h] = \max(C_j[h], C_i[h])$



# Vector clocks overview

- To compare to clocks $C_i$ and $C_j$
  - $C_i < C_j$ iff
    - $\forall h, C_i[h] \leq C_j[h]$ and

      $\exists h, C_i[h] < C_j[h]$



# Vector clocks overview

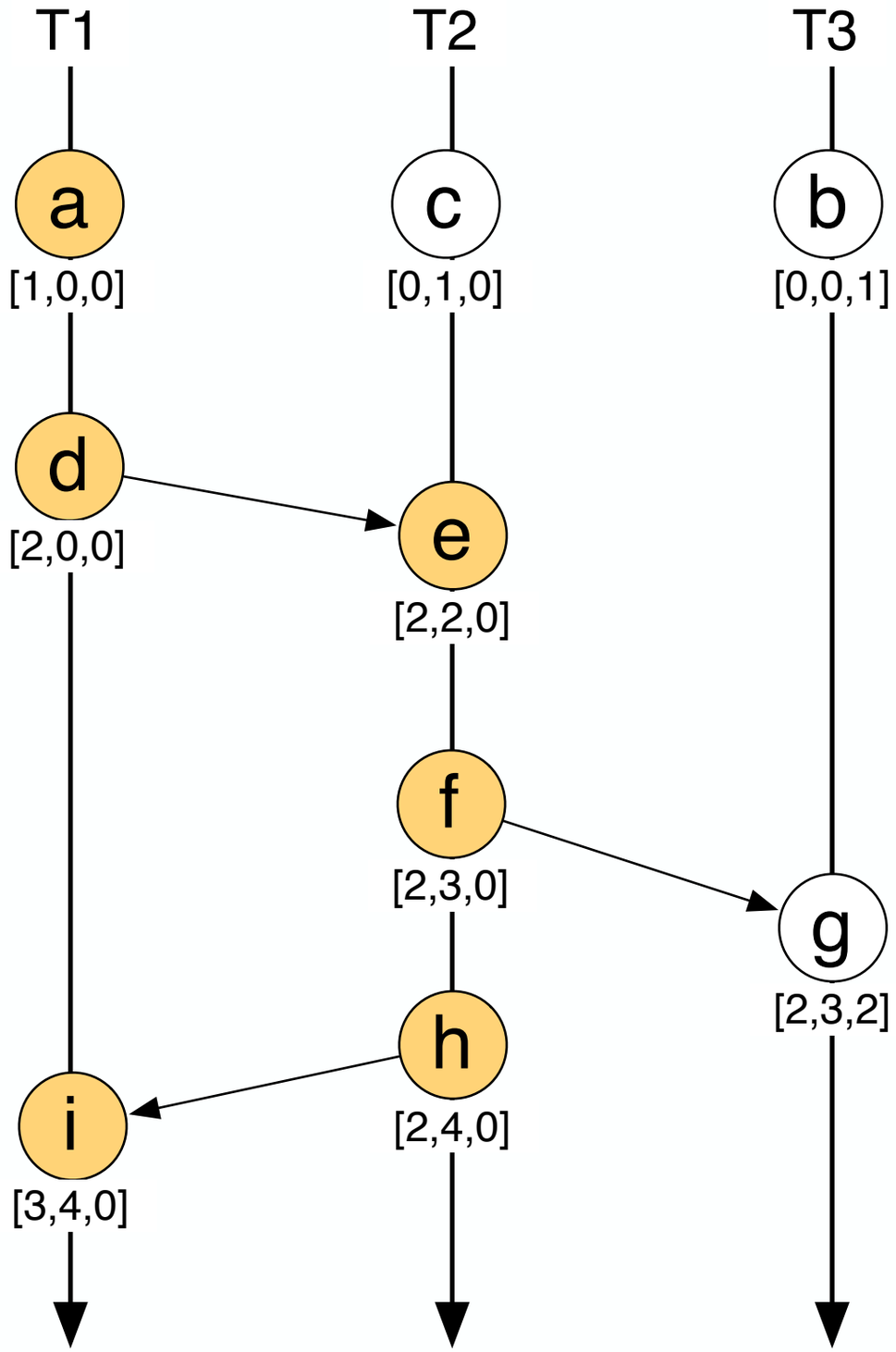
- To compare to clocks $C_i$ and $C_j$
  - $C_i < C_j$ iff
    - $\forall h, C_i[h] \leq C_j[h]$ and
    - $\exists h, C_i[h] < C_j[h]$

- Totally ordered events:

a < d < e < f < h < i

- Concurrent events:

b || c || d



# Vector clocks overview

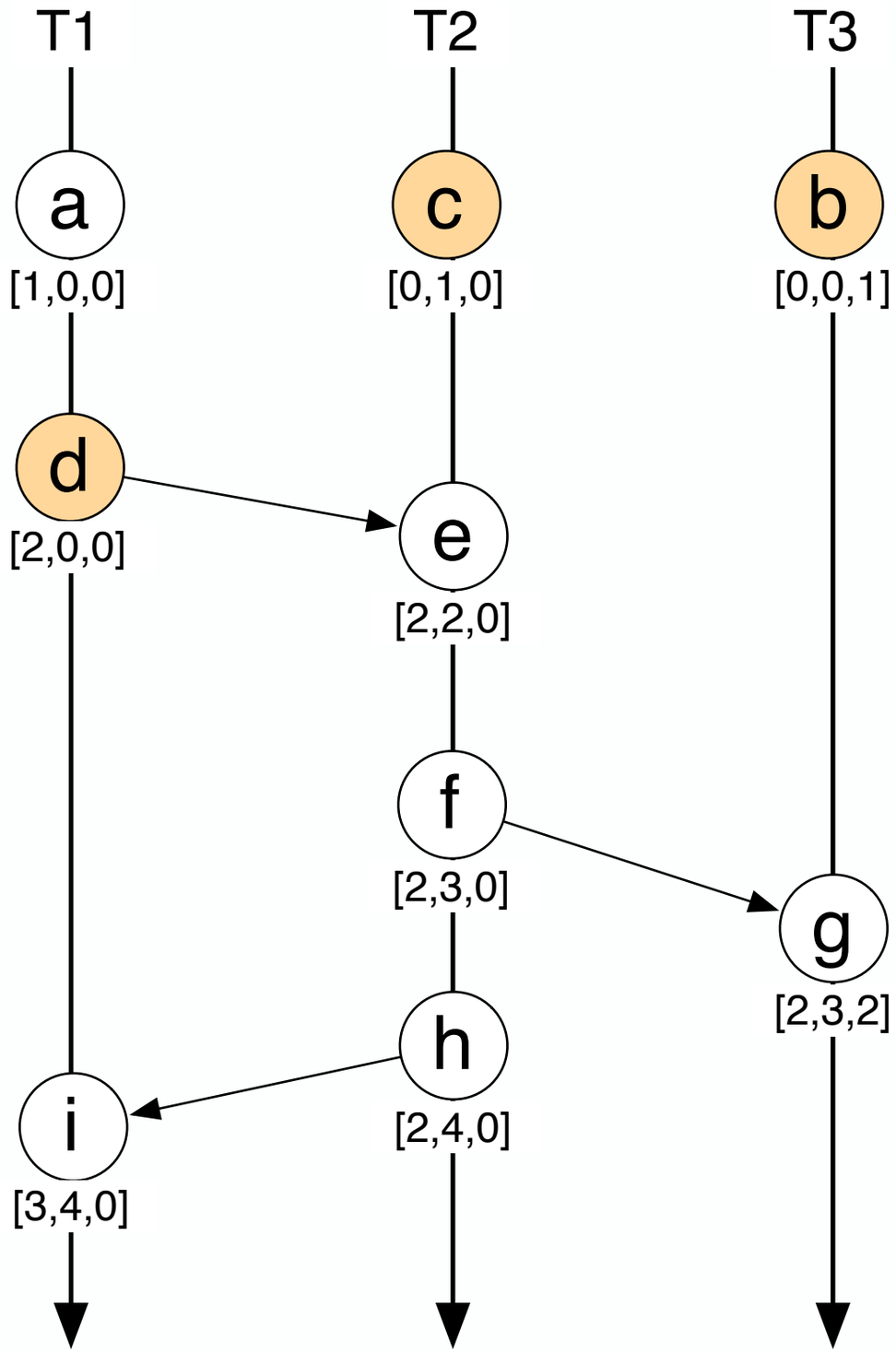
- To compare to clocks $C_i$ and $C_j$
  - $C_i < C_j$ iff
    - $\forall h, C_i[h] \leq C_j[h]$ and
    - $\exists h, C_i[h] < C_j[h]$

- Totally ordered events:

a < d < e < f < h < i

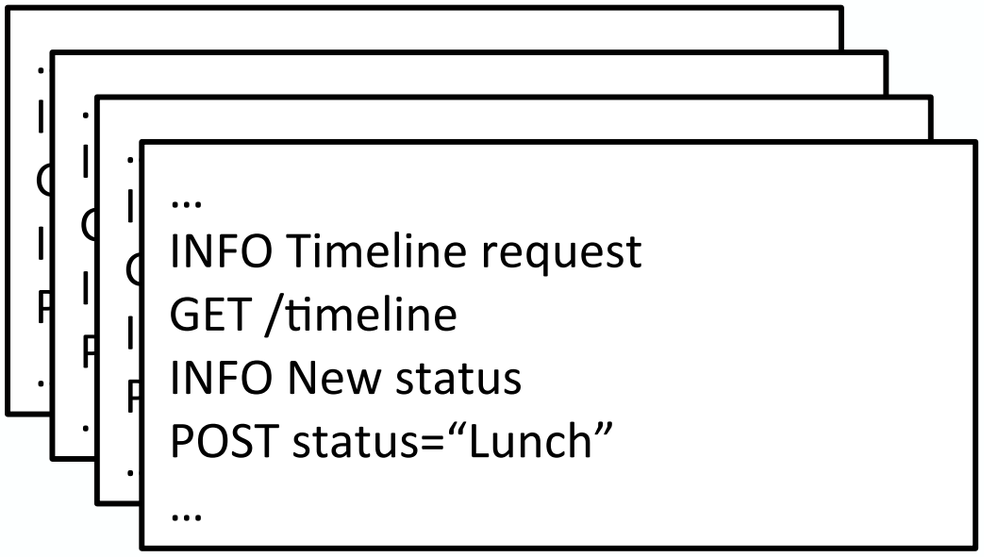
- Concurrent events:

b || c || d



# ShiVector instrumentation overview

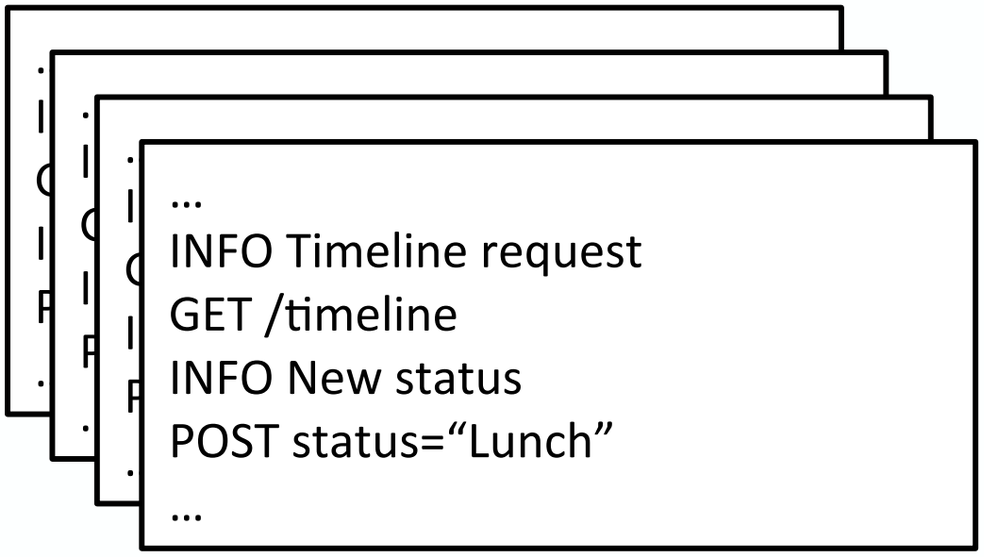
System



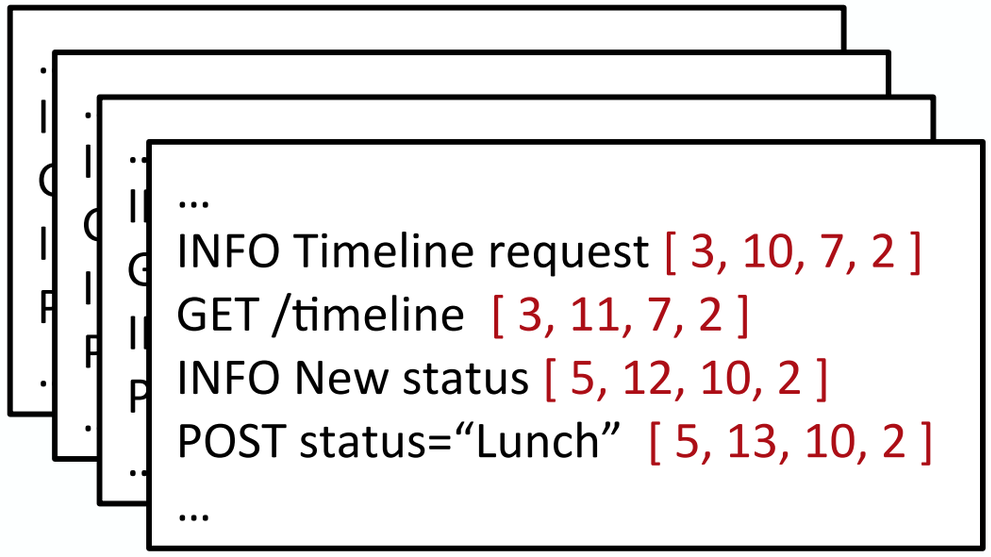

System + ShiVector

# ShiVector instrumentation overview

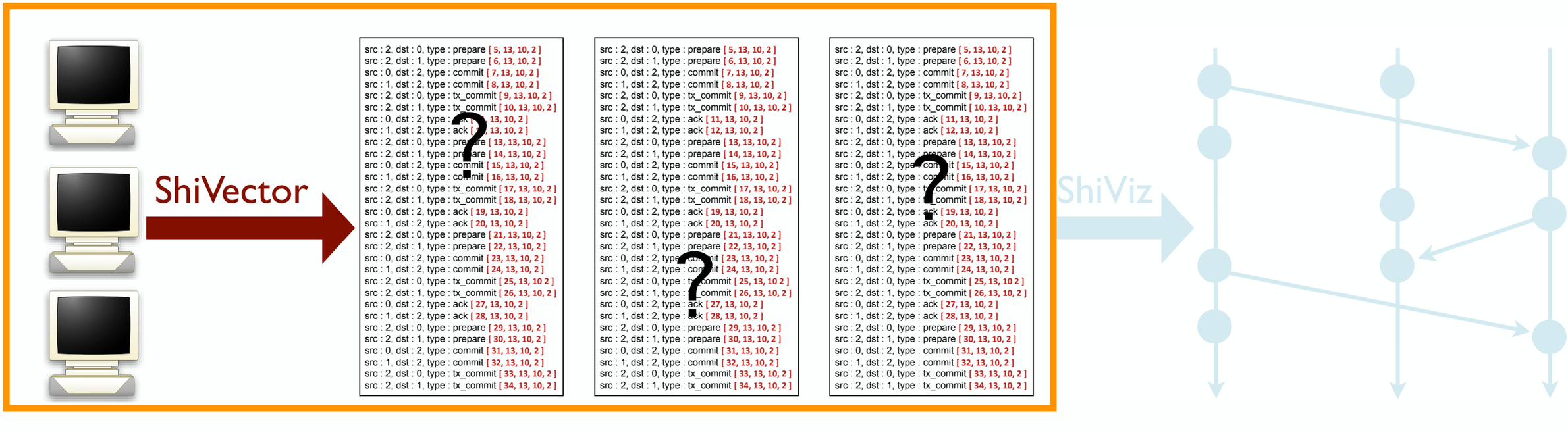
System



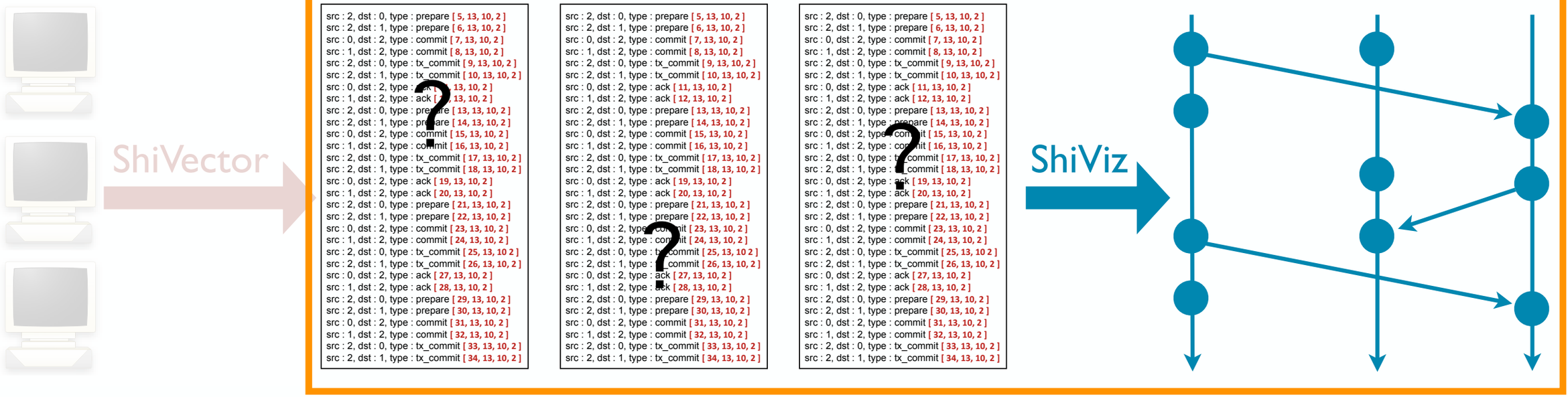

System + ShiVector



# Partial order instrumentation



# Partial order visualization

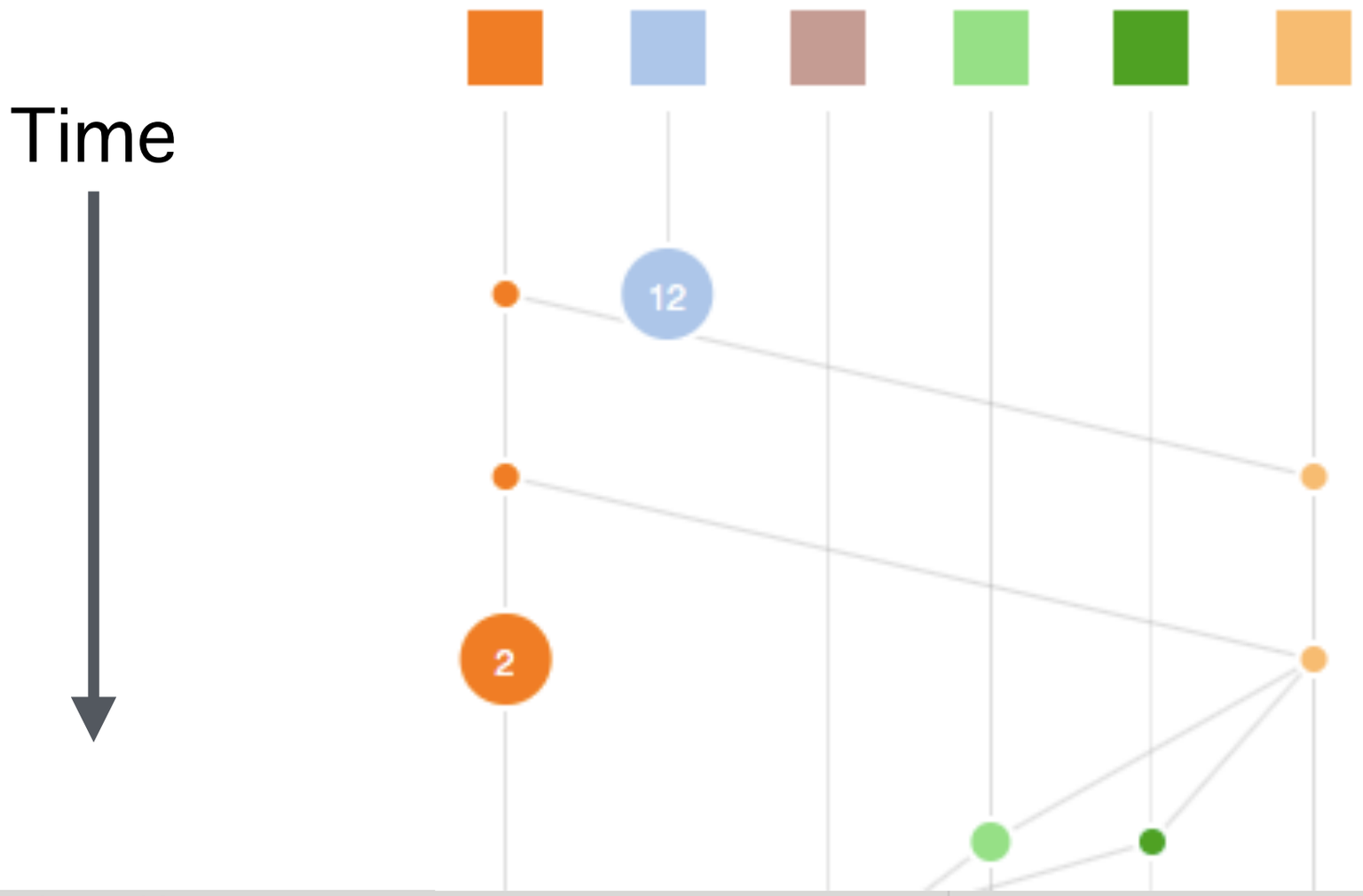


# ShiViz overview

- Take a log with partial timestamps as input, produce a communication graph as output
- Interactive tool to explore the partial order
- Targeted population: distributed system developers

# ShiViz: A distributed execution is a DAG

- Squares represent processes
- Circles represent process events
- Lines between events represent communication



# Demo

## ShiViz

**ShiViz** is a visualization engine that generates high level graphical summaries of distributed system execution logs.

Try out ShiViz below.

### Execution log

```
24.22.130.14 5/27/2013 10:53:39 AM GET /timeline uid=alice location=kansas
alice {"alice":1}
24.22.130.14 5/27/2013 10:53:48 AM INFO Timeline received: [] src=69.63.191.255
alice {"alice":2, "loadBalancer": 2, "eastDC":6, "westDC": 3}
24.22.130.14 5/27/2013 10:55:03 AM POST status="Breakfast" uid=alice location=kansas
alice {"alice":3, "loadBalancer": 2, "eastDC":6, "westDC": 3}
24.22.130.14 5/27/2013 10:55:11 AM INFO Status confirmed src=69.63.191.255
alice {"alice":4, "loadBalancer": 4, "eastDC":8, "westDC": 3}
24.22.130.14 5/27/2013 10:56:22 AM GET /timeline uid=alice location=kansas
alice {"alice":5, "loadBalancer": 4, "eastDC":8, "westDC": 3}
24.22.130.14 5/27/2013 10:56:24 AM INFO Timeline received: ["Breakfast"] src=69.63.191.255
alice {"alice":6, "loadBalancer": 6, "eastDC":12, "westDC": 6}
24.22.130.14 5/27/2013 10:59:28 AM POST status="Lunch" uid=alice location=kansas
alice {"alice":7, "loadBalancer": 6, "eastDC":12, "westDC": 6}
24.22.130.14 5/27/2013 10:59:52 AM INFO Status confirmed src=69.63.191.255
alice {"alice":8, "loadBalancer": 8, "eastDC":14, "westDC": 6}
24.22.130.14 5/27/2013 11:01:52 AM GET /timeline uid=alice location=kansas
alice {"alice":9, "loadBalancer": 8, "eastDC":14, "westDC": 6}
24.22.130.14 5/27/2013 11:01:59 AM INFO Timeline received: ["Breakfast"] src=204.15.23.252
alice {"alice":10, "loadBalancer": 10, "eastDC":14, "westDC": 8}
24.22.130.14 5/27/2013 11:02:13 AM POST error="Missing post" uid=alice location=kansas
alice {"alice":11, "loadBalancer": 10, "eastDC":14, "westDC": 8}
```

Examples: Facebook data-center log  Voldemort log  SimpleDB log

visualize

Browser options: Chrome (best), Firefox, or Safari (worst)

Does **not** work in IE

Avoid tablets and phones

# Begin assignment by browsing to:

# http://is.gd/etuhes

Questions? Raise your hand.