**Vector clocks**

\[ VC(e)[i] = \text{number of events executed by process } i \text{ that are causally related to } e \]
COMPARING VECTOR CLOCKS

Equality

\[ V = V' \equiv \forall k : 1 \leq k \leq n : V[k] = V'[k] \]
(i.e. all elements are the same)

Inequality

\[ V < V' \equiv (V \neq V') \land (\forall k : 1 \leq k \leq n : V[k] \leq V'[k]) \]

Examples: \([2,0,0] < [2,0,1] < [3,0,1] < [4,1,1]\)

Strong clock condition: \( p \rightarrow q \iff VC(p) < VC(q) \)
Comparing vector clocks

Strong clock condition: $p \rightarrow q \iff VC(p) < VC(q)$
CAUSAL DELIVERY

A “monitor” process wants to record all messages (e.g. deadlock detection, system snapshot, etc)

- Processes send copies of their messages to the monitor
- Only increment the local component of $VC$ for send events
Causal delivery rules

Monitor keeps an array $D$, where $D[i]$ is the number of messages delivered from process $i$. E.g. $[7, 1, 4]$

When is it OK to deliver a message $m$ with vector clock $VC(m)$?

Monitor delivers message $m$ from process $j$ when:

$D[j] = VC(m)[j] - 1$

$D[k] \geq VC(m)[k], \forall k \neq j$
Causal delivery

\[
D[j] = VC(m)[j] - 1
\]

\[
D[k] \geq VC(m)[k], \forall k \neq j
\]
Still missing a few pictures
Still missing a few people on Piazza
At capacity
Clock synchronization

What time is it?
Roman generals v2.0

Attack at midnight!

Chaaaaaaarge!

12:00am

ZZZZZZ

11:30pm
Clock drift

- **Bound on drift:** $\rho$

\[(1 - \rho)(t - t') \leq H(t) - H(t') \leq (1 + \rho)(t - t')\]

- $\rho$ is typically small ($10^{-6}$)

\[\rho^2 \approx 0\]

\[\frac{1}{1 - \rho} = 1 + \rho\]

\[\frac{1}{1 + \rho} = 1 - \rho\]
External vs internal synchronization

External Clock Synchronization:
keeps clock within some maximum deviation from an external time source.

- exchange of info about timing events of different systems
- can take actions at real-time deadlines

Internal Clock Synchronization:
keeps clocks within some maximum deviation from each other.

- can measure duration of distributed activities that start on one process and terminate on another
- can totally order events that occur in a distributed system
Probabilistic Clock Synchronization (Cristian)

- Master-Slave architecture
- Master can be connected to external time source
- Slaves read master's clock and adjust their own

How accurately can a slave read the master's clock?
Setup and assumptions

Goal: Synchronize the slave's clock with the master

Assume that minimum delay is known
Assume that clock drifts are known ($\rho$ for both)
The protocol

\[ t \text{ (real time)} \]

slave \( P(t) \)

master \( Q(t) \)

“time=?”

“time=\( T \)”

\[ t = x \]

\[ T \]

\[ Q(x) \]

Question: what is \( Q(x) \)?
Ideal scenario

$t$ (real time) $t = x$

slave $P(t)$ $Q(x) = T + \text{min}$ → Perfect synchronization!

master $Q(t)$

Assume no clock drift
Problem #1: message delay

\[ P(t) \quad Q(t) \]
\[ t \quad 2d \quad t \]
\[ Q(x) = T + \min \]
\[ \beta = 2d - 2\min \]

\[ P(t) \quad Q(t) \]
\[ t \quad 2d - \min \quad t \]
\[ Q(x) = T + 2d - \min \]
\[ \beta = 2d - 2\min \]

\[ P(t) \quad Q(t) \]
\[ t \quad 2d - \min \quad t \]
\[ Q(x) = T + \min \]
\[ \beta = 0 \]
Problem #2: slave drift

\[ 2d(1 - \rho) \leq 2D \leq 2d(1 + \rho) \]
Problem #3: master drift

During the master’s clock drifts
Even if you know $\beta$, there is still some uncertainty!
Cristian's algorithm

\[ \text{time} = \min + \alpha, \quad \min + \beta \]

\[ t = x \]

\[ \alpha, \beta \geq 0 \]

slave \[ P(t) \]

master \[ Q(t) \]

\[ 2D \]

\[ \text{time} = ? \]

\[ \text{time} = T \]

\[ T \]

\[ Q(x) = ? \]
Cristian’s algorithm

Naive estimation: \( Q(x) = T + (\text{min} + \beta) \)

(take master’s drift into account)

\[
Q(x) \in [T + (\text{min} + \beta)(1 - \rho), T + (\text{min} + \beta)(1 + \rho)]
\]

\[0 \leq \beta \leq 2d - 2\text{min}\] (take delay into account)

\[
Q(x) \in [T + (\text{min} + 0)(1 - \rho), T + (\text{min} + 2d - 2\text{min})(1 + \rho)]
= [T + (\text{min})(1 - \rho), T + (2d - \text{min})(1 + \rho)]
\]

(take slave’s drift into account)

\[2d \leq 2D(1 + \rho)\]

\[
Q(x) \in [T + (\text{min})(1 - \rho), T + (2D(1 + \rho) - \text{min})(1 + \rho)]
= [T + (\text{min})(1 - \rho), T + 2D(1 + 2\rho) - \text{min}(1 + \rho)]
\]
Slave's estimation and precision

Slave's best guess:  \( Q(x) = T + D(1 + 2\rho) - \min \cdot \rho \)

Maximum error:  \( e = D(1 + 2\rho) - \min \)

You can keep trying, until you achieve the required precision
(if that precision is reasonable)
Adjusting the clock

After synchronizing:

- If slave simply sets $P(x) = Q(x)$, it could create time discontinuities.
Adjusting the clock

Logical clock \[ C(t) = H(t) + A(t) \]

Hardware clock \quad Adjustment function

\[ C(t) = H(t) + A(t) \]