T3: Time and global state
TDT4190

Kjetil Sletten, Simen Skoglund and Christian Peter

Friday 7th March, 2014

1 Time

a)

Hver datamaskin har en *fysisk klokke*, hver klokke kalibrerer anderledes hver for seg. I et distribuert system, så vil ikke dette fungere med henhold til synkronisering. En *logisk klokke* er derfor nødvendig. En logisk klokke fungerer slik at den er uavhengig av den fysiske klokken ved at den oppdaterer en *program teller*. Denne telleren fungerer som en *timestamp* slik at andre prosesser over det distrubuerte systemet synkroniseres mht denne telleren. Hver prosess inkrementerer telleren før hver hendelse(Dette kan være 1 eller et annet positivt nummer).

b)

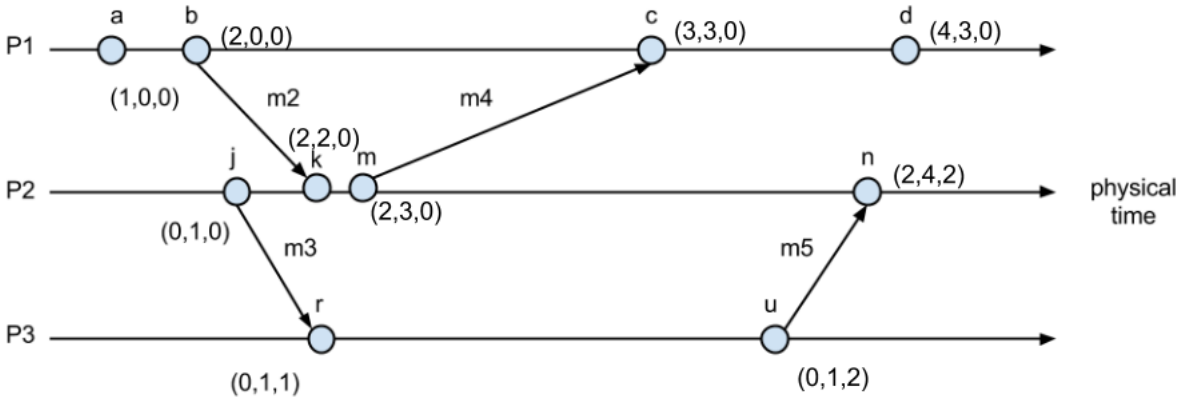
Happend-before er en relasjon mellom to hendelser. Når to hendelser skjer på samme prosess, vil hendelsen skje i forhold til den ordningen som prosessen ser dem. Ved sending av en melding til en annen prosess, vil sending forekomme som første

- $a -> j$ Nei
- $j -> c$ Ja, fordi regelen HB3, s. 623
- $k -> u$ Nei
- $a -> e$ Ja, fordi regelen HB3, s. 623

c)

Nei, når en hendelse e på en vilkårlig prosess sender en melding til hendelse f, så vet vi at Lamport timestamps for hendelse e er mindre enn hendelse f, altså $L(e) < L(f)$. Så hvis vi vet at $L(e) < L(f)$ så kan vi ikke si at e-¿f fordi vi ikke sikkert kan si om de henger sammen. Det eneste vi vet er at timestampen til e er mindre enn f. Derfor finnes vektor klokker, fordi å håndtere denne problematikken.

d)



e)

Bruker *21:46:01.003* da denne har den minste RTT(28 ms). Et enkelt estimat på tiden for når prosessen skal sette klokka, tatt fra side 618:

$$t + \frac{T_{round}}{2}$$

Dette gir oss en klokketid på *21:46:01.017*

Siden minstetid for å sende og motta en melding vil en sette klokken til:

$$\pm(\frac{28ms}{2} - 10ms) = \pm 4ms$$

Vi setter da klokken mellom *21:46:01.007* og *21:46:00.999*

2 NTP Synchronization

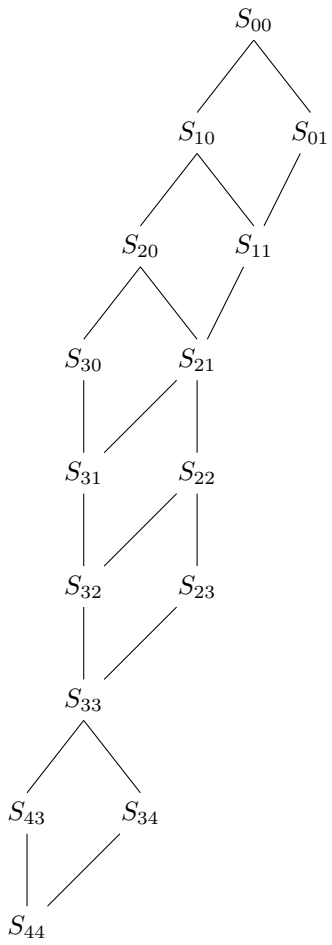
$T_{i-3} = 09:14:48.980$
$T_{i-2} = 09:14:59.030$
$T_{i-1} = 09:15:09.385$
$T_i = 09:15:19.425$

Tatt fra boken på s. 622 gir oss dette:

$$O_i = \frac{(T_{i-2} - T_{i-3} + T_{i-1} - T_i)}{2}$$

$$O_i = \frac{10050 - 10040}{2} = \frac{10}{2} = 5ms$$

3 Global State



4 Snapshot

Det som skjer når A starter snapshot algoritmen er:

1. A sender en melding med tilstand 21(for C1) til B.
2. A lagrer tilstanden sin til 21.
3. A sender en markørmelding til B.
4. B mottar markørmeldingen.
5. B finner ut at den nye tilstanden nå er 22.
6. B lagrer tilstanden som skjer når markørmeldingen er motatt.
7. B sender en markørmelding til A.
8. A mottar B sin markørmelding.
9. A lagrer C2 sin tilstand som nå er 22.