# Kapitel 3: Snapshots

19. december 2017 19:42

To forskellige:

- -Chandy-Lamport Algoritm
- -Lai-Yang algoritm

http://www.cs.vu.nl/~tcs/da/daslides.pdf side 26-39

5.2 Global Snapshot Algorithm 10:36

5.2 Global Snapshot Algorithm 10:36



Et øjebliksbillede af en udførelse af en distribueret algoritme skal returnere en konfiguration af en udførelse i samme beregning. Snapshots kan bruges til: Starte efter en fejl. Offline bestemmelse af stabile egenskaber, som forbliver sande, så snart de er blevet sande.

Eksempler: deadlock, garbage.

Debugging.

Vi skelner mellem basale meddelelser af den underliggende distribuerede algoritme og kontrol meddelelser af snapshot algoritmen.

Et øjebliksbillede af en (grundlæggende) udførelse består af:

et lokalt snapshot af (grundlæggende) tilstanden for hver proces, og

Kanaltilstanden for (grundlæggende) meddelelser i transit for hver kanal.

snapshot er meningsfuldt, hvis det er en konfiguration af en udførelse i samme beregning som den faktiske udførelse.

Vi skal undgå følgende situationer.

- 1. Process p tager et lokalt snapshot og sender derefter en besked m
- at behandle q, hvor:
- q tager et lokalt snapshot efter kvittering af m,

## Snapshots

Distributed Algoritmes bog side 24-27 med eksempler.

Billede af hvordan det distribuerede system ser ud lige nu. Bruger FIFO konceptet, medmindre det er Lai-Yang algoritmen.

M = signal Ø = Ingen beskeder

- eller m er inkluderet i kanaltilstanden pg.
- 2. p sender m til q, og tager derefter et lokalt øjebliksbillede, hvor:
- q tager et lokalt øjebliksbillede inden kvitteringen af m,
- og m er ikke inkluderet i kanaltilstanden pg.

### **Chandy-Lamport algorithm:**

Overvej et rettet netværk med FIFO-kanaler.

Initiativtagerne tager et lokalt øjebliksbillede af deres tilstand og sender en kontrolmeddelelse hmarkeri til deres naboer.

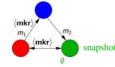
Når en proces, der endnu ikke har taget et øjebliksbillede, modtager hmarkeri, er det Jeg tager et lokalt øjebliksbillede af sin stat, og

Jeg sender hmarkeri til alle sine naboer.

Proces q beregner som kanaltilstand for pq de meddelelser, den modtager via pq efter at have taget sit lokale øjebliksbillede og før han modtog hmarkeri fra s. Hvis kanalerne er FIFO, giver dette et meningsfuldt øjebliksbillede.

Meddelelseskompleksitet: Θ (E) Værste tilfælde kompleksitet: Ο (D)







Snapshot (behandler rød / blå / grøn, kanaler Ø, Ø, Ø, {m2}) er ikke en konfiguration i den faktiske udførelse.
Sendet af m1 er ikke kausal før sendet af m2.
Så snapshot er en konfiguration af en eksekvering, der er i den samme beregning som den faktiske udførelse:



Påstand: Hvis en post-snapshot hændelse e er kausal før et arrangement f, så er f også et post-snapshot.

Dette indebærer, at snapshotet er en konfiguration af en udførelse

det er i samme beregning som den faktiske udførelse. Bevis: Den sag, at e og f forekommer ved samme proces er trivielt.

Lad os være en send og f den tilsvarende modtagelsesbegivenhed.

Lad os forekomme ved p og f ved q.

e er post-snapshot på p, så p sendte hmarkeri til q før e.

Kanaler er FIFO, så q modtager denne hmarkeri før f.

Dermed f efter-snapshot ved q.

#### Lai-Yang algorithm

Antag, at kanaler ikke er FIFO. Vi bruger piggybacking. Initiativtagerne tager et lokalt øjebliksbillede af deres tilstand. Når en proces har taget sit lokale øjebliksbillede, tilføjer det rigtigt til hver udgående grundlæggende besked.

Når en proces, der endnu ikke har taget et øjebliksbillede, modtager en besked med ægte eller en kontrolmeddelelse (se næste dias) for første gang, det tager et lokalt øjebliksbillede af sin tilstand inden modtagelsen af denne meddelelse. Proces q beregner som kanaltilstand for pq de grundlæggende meddelelser uden mærket sandt, at det modtager via pg efter sit lokale øjebliksbillede.

Spørgsmål: Hvordan kan g vide, hvornår den kan bestemme kanaltilstanden af pq?

p sender en kontrolmeddelelse til q, der oplyser q hvor mange grundlæggende meddelelser uden mærket sande p sendt til pq.

Disse kontrolmeddelelser sikrer også, at alle processer til sidst tager et lokalt øjebliksbillede.

Spørgsmål: Hvordan kan flere efterfølgende snapshots understøttes? Svar: Hvert øjebliksbillede er forsynet med et sekvensnummer. Grundlæggende besked bærer sekvensnummeret for det sidste snapshot på afsenderen (i stedet for sand eller falsk). Kontrol meddelelser bærer sekvensnummeret på deres øjebliksbillede.

#### Referat

fully asynchronous message passing framework channels are non-FIFO, and can be directed or undirected configurations and transitions at the global level states and events (internal/send/receive) at local leve (non)initiator (de)centralized algorithm; causal order ≺ on events in an execution computation of executions, by reordering concurrent events snapshot algorithm to compute a configuration of a computation basic/control algorithm