Kapitel 5: Deadlock Detection

21. december 2017 13:26

To forskellige:
Wait-for graphs
Bracha-Toueg algoritm

http://www.cs.vu.nl/~tcs/da/daslides.pdf side 62-89

En deadlock opstår, hvis der er en procescyklus, der venter til:

- En anden proces på cyklen sender noget input (kommunikation deadlock)
- eller ressourcer fra andre processer på cyklen frigives (ressourceophæng)

Begge typer deadlock fanges af N-out-of-M-modellen:
En proces kan vente på, at N tildeler ud af M-anmodninger.

Eksempler:

- En proces venter på en besked fra en gruppe processer: N = 1
- En database transaktion skal først låses flere filer: N = M.

Wait-for graphs
En (ikke-blokeret) proces kan udstede en anmodning om M andre processer, og bliver blokeret, indtil N af disse anmodninger er blevet tildelt.
Derefter informerer den de resterende M-N processer om anmodningen kan afskediges.

Kun ikke-blokerede processer kan give en anmodning.

En (rettet) ventetid indfanger afhængigheder mellem processer.

Der er en kant fra node p til node q, hvis p sendte en anmodning til q
det var endnu ikke afvist af p eller givet af q.

Eksempel:
Antag at proces p skal vente på en besked fra proces q.

I ventetidskurven sender node p en forespørgsel til node q.
Derefter oprettes edge pq i wait-for-graph, og p bliver blokeret.

Når q sender en besked til p, er anmodningen fra p givet.

Derefter fjernes kant pq fra wait-for-graph, og p bliver
frigives.

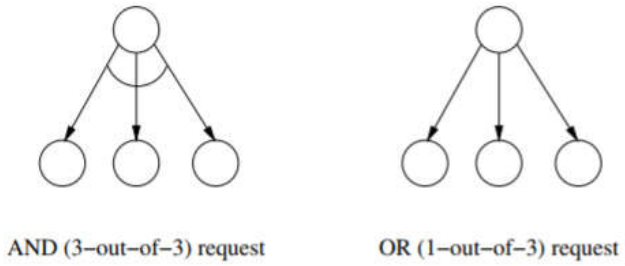
Eksempel 2:
Antag to processer p og q vil kræve en ressource.
I wait-for-graph, nodes u, v der repræsenterer p, q sender en anmodning til noden w repræsenterer ressourcen. Edges jou og vw oprettes.

Da ressourcen er fri, gives ressourcen til at sige p.
Så w sender et tilskud til dig. Edge uw er fjernet.

Den grundlæggende (gensidige udelukkelses) algoritme kræver, at ressourcen skal frigives ved p, før q kan hævde det.
Så w sender en forespørgsel til u, der skaber edge wu i wait-for-graph.

Når p frigiver ressourcen, giver du anmodningen fra w.
Edge wu fjernes.

Resursen gives til q. Derfor tildeler w anmodningen fra v.
Edge vw fjernes, og edge wv oprettes.

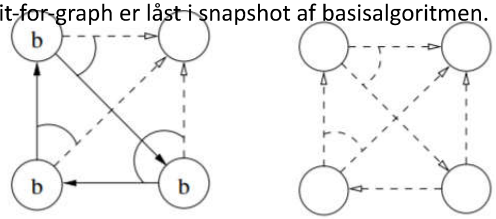


AND (3–out–of–3) request OR (1–out–of–3) request

Et snapshot tages af wait-for-graph.
En statisk analyse på ventetidskurven kan afsløre deadlocks:

- ikke-blokerede knudepunkter kan give anmodninger.
- Når en anmodning gives, fjernes den tilsvarende kant.
- Når en N-out-of-M-anmodning har modtaget N, bliver requesten ublockeret.
  (De resterende M - N udgående edges afvises.)

Når der ikke er flere tilskud, kan knudepunkter forblive blokeret i
Wait-for-graph er låst i snapshot af basisalgoritmen.



Deadlock No deadlock

Bracha-Toueg algoritm
Givet et ikke-direkte netværk og en grundlæggende algoritme.
En proces, som mistænker det er deadlock, initierer
et (Lai-Yang) snapshot for at beregne wait-for-graph.

Hver node u tager et lokalt Snapshot af:

- anmoder det sendt eller modtaget, som endnu ikke er givet eller afskediget;
- tildele og afvise beskeder i kanter.

Så beregner det:
Out(u): de noder, den sendte en anmodning om (ikke tildelt)
In(u): noderne modtog en anmodning fra (ikke afskediget)

Kunne vi anvende Bracha-Toueg-algoritmen til sig selv for at etablere
at det er en deadlock-fri algoritme?
Svar: Nej.
Bracha-Toueg-algoritmen kan kun fastslå om en deadlock, som er til stede i et snapshot af en beregning
af basisalgoritmen.

Resten:
Side 76 til 89