

# dConc Aflevering 5 - DA6

Christian Zhuang-Qing Nielsen

201504624, christian@csn.dk

December 10, 2016

*Introduktion og trin 1-5 er fra aflevering 3. Trin 6-7 er nye.*

## Introduktion

Pointen i denne opgave er at se hvorvidt `notifyAll()` er skidt for concurrent programmer. Som eksempel bliver Producer-Buffer-Consumer fra uge 1 brugt. Rapporten er opdelt i fem trin hvor svarene på disse følger her:

### Trin 1

Jeg har tilføjet try/catches der hvor bufferen kaster `InterruptedExceptions`.

### Trin 2

Hvis hver producer kun laver en enkelt meddelelse og consumer, så vil `wait()` blive kaldt to gange for hver producer (hver gang den skal frem og tilbage). Derfor vil jeg mene at tidskompleksiteten af dette i værste tilfælde er  $2n(n-1)/2 = O(n^2)$ , eftersom der højst er 2 while-loops der venter på hinanden på et givent tidspunkt. `wait()` udgør et problem når kommer til at blive udsat for starvation, samt det høje cpu-brug ved gentagne context-switches. Desuden er der en del tidsbrug ved at skulle vække alle trådene på én gang og så derefter sætte dem alle på nær én til at vente derefter.

### Trin 3

Her har jeg skrevet en ny main metode som tager et integer input som så er antallet af producers den skal skabe. Consumer målen tidsforbruget ved bare at printe hver gang den får en ny streng. Se den lille linje på grafen figur på side 3, der også bekræfter  $O(n^2)$  tidsforbruget.

## Trin 4

Eftersom at trådenene ikke har brug for at blive vækket i en bestemt rækkefølge, så er det helt fint bare at vække dem alle på én gang med `notifyAll()`. Når vi bare bruger `notify()` er det sværere at få vækket de relevante tråde.

## Trin 5

Hvis vi havde fået givet en magisk `notify()` af Julemanden, som kun vækkede relevante tråde op hver gang, kunne vi spare hele trinnet med at vække en masse threads og så sætte dem til at sove igen med det samme. På den måde kunne vi få tiden reduceret til lineært  $O(n)$  frem for polynomisk.

## Trin 6

Jeg har omskrevet bufferen således at den bruger semaforer (der er opbygget med `notify()` frem for `notifyAll()`). Eftersom vi kun vækker de relevante tråde, og at der er  $n$  tråde, så vil tidskompleksiteten være  $O(2n) = O(n)$ , altså lineær.

## Trin 7

Tidsforbruget er i magen til `notifyAll()` i starten, men stikker af og er markant bedre når antallet af producers når over 1000. Se vedhæftet fil for tidsforbrug og grafen figur på side 3 i bilaget. Den blå linje er den nye graf med semaforer og den lilla er den gamle. Det fremgår ikke ligetil, at tidsforbruget er lineært, men dette er tydeligt hvis man sætter graf-trinene er manuelt sat til at være ens på både x-akse og y-akse.

## Bilag

Time required in ms for the amount of producers using semaphores

