

Tweede Deeltoets Concurrency

7 november 2014, 11.00–13.00, Educ-Γ.

Maak vragen 1 en 2 op pagina 1, 3 en 4 op pagina 2, en 5 en 6 op pagina 3. Vragen 1 t/m 4 zijn 3pt, vraag 5 is 4pt en vraag 6 is 2pt. T2 is totaal plus 1, gedeeld door 1,8.

- Minimum leiders bij Flexibele Electie:** Bij Flexibele Electie voor n processen en robuustheid t , doet elk proces (1) een *shout* van zijn id, (2) *collect* $n - t$ id's (waaronder zijn eigen), en (3) beslist *leider* te worden als zijn id tot de $t + 1$ kleinste behoort van degenen die hij ziet. Bewijs dat tenminste èèn correct proces *leider* wordt.
- Ronden herhalen:** Steven wil dat zijn drie servers elk een *verschillend* identificatienummer krijgen van slechts twee bits. Een *ronde* bestaat uit een random keuze van een nummer door elke server, en een check of de drie keuzen verschillend zijn. Ronden worden net zolang herhaald totdat het gewenste doel bereikt is.
 - Bewijs, dat elke ronde die begint, kans *meer dan* $1/4$ heeft om het doel te bereiken.
 - Wat is de kans dat het algoritme na k ronden nog niet is getermineerd?
 - Wat is de (verwachte) complexiteit? Is dit een Monte Carlo of een Las Vegas algoritme? Leg uit!
- Omslagpunt work en span:** Voor zekere taak heb je een parallel algoritme A1 met work w_1 en span s_1 . Er is een alternatief algoritme A2 met slechter work $w_2 > w_1$ maar betere span $s_2 < s_1$. Voor welke aantallen cores, p , verwacht je van A2 een lagere rekentijd dan van A1? (Geef je antwoord asymptotisch, dus in grote- O of Θ .)
- Ben-Or en meerderheid:** Een ronde in Ben-Or's algoritme begint met het *shouten* van een waarde (0 of 1) en een *collect* van $n - t$ waarden. Als een station meer dan $n/2$ keer dezelfde waarde ziet, zal het een *propose* daarvoor rondsturen. Als een station meer dan t proposes voor een waarde ziet, zal het beslissen op die waarde.
 - Stel dat *meer dan* $n/2 + t$ processen een ronde beginnen met dezelfde waarde v ; is een beslissing voor v *zeker* in die ronde?
 - Stel dat *precies* $n/2 + t$ processen een ronde beginnen met dezelfde waarde v ; is een beslissing voor v *mogelijk* in die ronde?
 - Waarom werkt het algoritme sneller naarmate t kleiner is?
- Parallele Eenummers:** Gegeven een reeks A van nullen en enen. Het *eennummer* van een positie met waarde 1 geeft aan, de hoeveelste opeenvolgende 1 het is. Een positie met waarde 0 heeft eennummer 0. Voor 11011011111000111 zijn de eennummers 12012012345000123. Piets Recursieve Eenummeraar berekent eerst de PRE van het linker- en rechterdeel:

```
PRE(A,p,r) { // Zet A[p,...,r-1] om naar eennummers
  q=(p+r)/2 ; invoke PRE(A,p,q) en PRE(A,q,r) ;
```

 - Hoe ziet de voorbeeldrij er hierna uit, en wat moet er dan nog gebeuren?
 - Maak Piets programma af.
 - Analyseer work en span (met de Master Theorem).
 - Is Piets algoritme *efficient* en *optimaal*?
- Welordening:** (a) Wanneer is een verzameling *welgeordend*? Geef de definitie.
(b) Zij A de verzameling van strings over het alfabet $\{a, b, c\}$, met hierop de lexicografische ordening. Is A welgeordend?