

Tweede Toets Concurrency

2 februari 2017, 8.30 – 10.30, Educ- β .

Motiveer je antwoorden *kort!* Zet je mobiel uit. Stel geen vragen over deze toets; als je een vraag niet duidelijk vindt, schrijf dan op hoe je de vraag interpreteert en beantwoord de vraag zoals je hem begrijpt.

Cijfer: Elke *deelvraag* telt even zwaar. Vragen 1, 3 en 5 zijn 2pt, vragen 2 en 4 zijn 3pt en vraag 6 is 4pt. Te halen 16, T2 is totaal plus 1 gedeeld door 1,6.

- Inverse Prefix Sum:** Gegeven is een array B . Gevraagd wordt een array A te berekenen, zodanig dat B de PrefixSum is van A .
 - Geef een zo goed mogelijk parallel algoritme hiervoor.
 - Wat zijn work en span van je algoritme?
- Work en Span vergelijken:** Arnold heeft een parallel algoritme ontworpen met *work* $\Theta(n^2)$ en *span* $\Theta(\lg^2 n)$. Bert zegt dat zijn eigen algoritme voor deze taak beter is. Bert heeft weliswaar een hogere span, namelijk $\Theta(\lg^3 n)$, maar zijn work is lager, namelijk $\Theta(n^{1.5})$.
 - Welk algoritme zal beter presteren bij een lineair aantal, dus $\Theta(n)$, processors? Leg uit!
 - Welk algoritme zal beter presteren bij een kwadratisch aantal, dus $\Theta(n^2)$, processors? Leg uit!
 - Voor welke aantallen van processors is Arnolds algoritme sneller?
- Heterogeen:** In een heterogeen systeem werken CPU en GPU samen om een probleem op te lossen.
 - In bepaalde gevallen kan code die sneller door een GPU uitgevoerd kan worden toch beter op de CPU uitgevoerd worden. Leg uit in welke gevallen, en waarom.
 - In een hypothetische heterogene renderer wordt het beeld opgedeeld in tiles. CPU en GPU gebruiken work stealing om deze set tiles te renderen. Waarom is dit een slecht idee?
- Parallel Reduction:** In de slides wordt uitgelegd dat voor een parallel reduction de combiner functie associatief moet zijn.
 - Leg uit waarom de combiner functie associatief moet zijn.
 - Laat zien dat floating point addition niet associatief is.
 - Laat zien hoe parallel reduction efficiënt met vector operaties uitgevoerd kan worden. Leg uit waarom de combiner functie bij gebruik van vector operaties, naast associatief ook commutatief moet zijn.
- Monte Carlo:** Deze toets bestaat uit zes opgaven.
 - Leg uit waarom deze vorm van toetsen gezien kan worden als een Monte-Carlo proces. Wat is de integraal in dit verband?
 - Is er sprake van stratificatie?

6. **Parallele Plip-King:** Plipper is een nieuw sociaal netwerk waar deelnemers maar 1 ding kunnen doen namelijk andere deelnemers *plippen* (je kunt *niet* jezelf plippen). In een groep A van deelnemers is x een *Plip-King* als hij door alle anderen in A geplipt is, maar zelf niemand in A plipt. Een verzameling van $n \geq 2$ deelnemers kan nul of een King bevatten, maar niet meer. Hoewel de informatie over onderling plipgedrag van n deelnemers in totaal bijna n^2 bits bevat, is het mogelijk om sequentieel in lineaire (dwz., $O(n)$) tijd te bepalen of die groep een Plip-King heeft.

Deze opdracht kijkt naar een *parallel Plip-King algoritme*. Voor identifiers i en j kun je in $O(1)$ (dus constante) tijd opvragen of i j plipt. Om te bepalen of groep A van n personen een King heeft, splitsen we A in twee groepen A_1 en A_2 , en bepalen recursief of die een Plip-King hebben.

(a) Als A een King heeft, dan moet dat de King van A_1 of A_2 zijn. Wat moet je doen om te controleren of de deel-Kings inderdaad King van A zijn?

(b) Stel vergelijkingen op voor de Work en Span van dit algoritme.

(c) Wat is de uitkomst van de Work en Span?

(d) Is dit algoritme efficient? Is het optimaal?