

Algoritmen voor parallele computers

Examen 29 januari 2004 (voormiddag)

Voor de vragen 1 - 4 is een maximale voorbereidingstijd van 1u 30min voorzien. Hierna wordt over deze vragen mondeling examen afgelegd. Gebruik gerust uw cursusnota's om uw antwoord mondeling toe te lichten (dit is een open-boek-examen!). Vervolgens kan je gedurende maximaal 1 uur verderwerken aan vraag 5. Hierna wordt over deze vraag mondeling examen afgelegd.

1. De optimale implementatie van de communicatie-bewerking 'allen-naar-allen gepersonaliseerde verzending' op een hyperkubus-architectuur met 'cut-through' routing hangt af van de lengte van de boodschap. Leg uit en bespreek de verschillende algoritmen. Wat is de uitvoeringstijd i.f.v. de boodschaplengte m en het aantal processoren p ?
2. Sorteren op een parallele computer met verspreid geheugen ('message passing'). Leg uit hoe het 'oneven-even transpositie-algoritme' werkt, en bespreek de parallele efficiëntie en de iso-efficiëntiefunctie. Dit algoritme is een variante van 'bubble-sort', zoals rood-zwart Gauss-Seidel-iteratie een variante is van de klassieke (lexicografische) Gauss-Seidel-iteratie voor het oplossen van stelsels afkomstig van roosterproblemen. Verklaar het verband.
3. Beantwoord beknopt de volgende vragen:
 - Stel dat je een zeer efficiënt parallel algoritme hebt ontwikkeld voor een hyperkubus-architectuur, maar je beschikt enkel over een parallele computer met een 2D rooster-topologie. Hoe kan je het hyperkubus-algoritme best implementeren op de 2D rooster-topologie?
 - Wat is de bisectiebandbreedte van een parallele computer? Wat kan je hieruit afleiden?
4. Bespreking van de gemaakte samenvatting over parallele ray-tracing [Chalmers] & [Reinhard] of over parallele data-mining [Joshi].

Vraag 5: zie apart blz

5. Roosterproblemen op een parallelle computer met verspreid geheugen.

Veronderstel dat een beeld van 1024×1024 pixels bloksgewijs gepartitioneerd is over de geheugens van 32 processoren, die een 2D-processorenrooster vormen (8×4 processoren). Veronderstel dat een beeldverwerkingsalgoritme wordt uitgevoerd waarbij elke pixelwaarde wordt vervangen door het gemiddelde van zichzelf en zijn 24 burens (25-punts-rekenmolecule), zie figuur.

Hoe zal je de communicatie programmeren met asynchrone blokkerende communicatieprimitieven? (enkel 'hoog niveau' beschrijving gevraagd)

Veronderstel dat de opstarttijd t_s voor communicatie gelijk is aan $5000t_{calc}$ en de (marginale) communicatietijd per woord $t_w = 20t_{calc}$, waarbij t_{calc} de tijd is voor 1 rekenkundige bewerking.

Bereken de parallelle efficiëntie.

Wat verandert er aan deze berekening indien het beeld slechts 1022×1022 pixels bevat?