

Oppgave 3 (vekt 20%)

I Oblig-1 har vi sett på “image denoising”, hvor beregningen på hver piksel er på følgende form:

$$\bar{u}_{i,j} = u_{i,j} + \kappa (u_{i-1,j} + u_{i,j-1} - 4u_{i,j} + u_{i,j+1} + u_{i+1,j}).$$

La oss si at MPI brukes til å lage en parallell implementasjon av “image denoising”, og vi antar at `MPI_Send` og `MPI_Recv` brukes til å utveksle data mellom naboene. Videre antar vi at tiden det tar å utveksle en melding av størrelse m mellom to MPI prosesser er som

$$t_s + t_w m,$$

hvor t_s og t_w er to kjente konstantverdier.

For et bildet med $n \times n$ piksler og P MPI prosesser, diskuter når det lønner seg å bruke en 2D blokk-partisjonering fremfor en 1D blokk-partisjonering. (Hint: vi er på jakt etter en relasjon mellom n , t_s , t_w og P .)

Oppgave 4 (vekt 25%)

Vi ønsker å beregne $y = Ax$, hvor A er en $n \times n$ matrise, og x og y er to vektorer av lengden n .

4a (vekt 10%)

Forklar hvordan matrise-vektor produkten kan paralleliseres, dersom vi antar en 1D blokk-partisjonering av A , x og y .

4b (vekt 15%)

Ifølge læreboka vil tidsbruken til ovenstående parallelisering være

$$T_P = \frac{n^2}{p} + t_s \log p + t_w n$$

hvor p er antall prosessorer, t_s og t_w er to kjente konstantverdier.

Gjør en skaleringsanalyse (scalability analysis) ved hjelp av “isoefficiency”-metrikken.

Oppgave 5 (vekt 30%)

Problemstilling “all-pairs shortest paths” handler om å finne den korteste vei mellom to og to noder i en graf. Som utgangspunkt har vi en matrise A som gir en oversikt over de direkte veiene i grafen. Som resultat vil vi generere en matrise D , hvor $d_{i,j}$ inneholder lengden til den korteste veien fra node i til node j .

(Fortsettes på side 3.)

5a (vekt 10%)

Beregn D dersom A er som følgende

$$\begin{bmatrix} 0 & 4 & \infty & \infty \\ 2 & 0 & 3 & 3 \\ \infty & 4 & 0 & 3 \\ \infty & 2 & 4 & 0 \end{bmatrix}$$

5b (vekt 10%)

Forklar Floyds algoritme som kan brukes til å løse den generelle problemstillingen.

5c (vekt 10%)

Implementer en parallell versjon av Floyds algoritme ved hjelp av OpenMP. (Antar at matrisen A er input og det er totalt n noder i grafen.)