UMEÅ UNIVERSITET
Institutionen för datavetenskap
Mikael Rännar (Erik Elmroth,
Robert Granat och Bo Kågström)

Tentamen på kursen

Design och analys av algoritmer för parallelldatorsystem

onsdagen den 24 augusti 2005 kl 9 - 15 i Skrivsal 6

Tentamen kan maximalt ge 40 poäng. För betyget 3 krävs 20 poäng, för betyget 4 krävs 26 poäng och för betyget 5 krävs 32 poäng. Poängtal anges inom parentes för varje uppgift. Uppgifterna är ej placerade i svårighetsordning.
Hjälpmedel:

- Miniräknare

Skrivsalen får lämnas tidigast en halvtimme efter att skrivningen påbörjats.

OBS!!!
Skriv namn på varje blad och börja varje uppgift på nytt blad. Var noga med att redovisa alla beräkningar. Markera på försättsbladet vilka uppgifter du lämnar in lösning till och lägg lösningarna i nummerordning.
Skriv TYDLIGT!

Lycka Till!

Uppgift 1 *Skalbarhetsanalys* (2+1+2+2+2=9p)
Kostnaden (mätt i tid) för en sekventiell algoritm för ett givet problem är $T_1 = n^3$. En parallell motsvarighet till algoritmen har kostnaden $T_p = n^3/p + \sqrt{p}nt_s + n^2t_w$ då den exekverar på p processorer.

a. Beräkna explicita uttryck för den parallella algoritmens uppsnabbning (*parallel speedup*) och effektivitet (*parallel efficiency*), samt avgör om (och i så fall under vilka förutsättningar) algoritmen är kostnadsoptimal (*cost optimal*).

b. Förklara vad som avses med begreppet skalbarhet (*scalability*)? Vad säger isoeffektivitetsfunktionen (*iso-efficiency function*) om en algoritms skalbarhet?

c. Beräkna ett explicit uttryck för algoritmens asymptotiska isoeffektivitetsfunktion.

d. Antag att $t_s = 1000$ och $t_w = 10$. Beräkna med fyra decimalers noggrannhet effektiviteten då $p = 10$ och $n = 1000$.

Hur stort ska, enligt isoeffektivitetsfunktionen, problemet vara för att samma effektivitet ska erhållas för tio gånger så många processorer, dvs $\tilde{p} = 100$. Ange ditt svar som storlek för aktuellt n-värde, betecknat $\tilde{n}$.

e. Beräkna med samma antagande som ovan effektiviteten för $\tilde{p} = 100$ processorer och det värde på $\tilde{n}$ du beräknat ovan. Ange effektiviteten med två korrekta decimaler.

Vad kan sägas om detta värde jämfört med det som beräknades i ovanstående deluppgift. Förklara eventuella skillnader.

Vad kan sägas om utvecklingen av effektiviteten för ytterliggare större processorantal (och problemstorlekar skalade i enlighet med isoeffektivitetsfunktionen).

Uppgift 2 *Biblioteksprogramvara* (3+2+2=7p)
I denna uppgift skall du berätta vad du känner till om programvarubiblioteket ScaLAPACK.

a. Redogör för *mjukvaruhierarkin* i ScaLAPACK. Var noga med att skilja på komponenter som opererar *globalt* respektive *lokalt*. Beskriv kortfattat vad de olika komponenterna innehåller för typ av rutiner.

b. Grundläggande i ScaLAPACK är att man endast har en typ av *processortopologi* och ett enhetligt sätt att *distribuera data*. Vilken topologi och vilken typ av datadistribution avses?

c. I ScaLAPACK utnyttjas en *objektorienterad ansats* för att beskriva hur datastrukturerna är distribuerade över de ingående processorerna. Beskriv hur detta är konstruerat.

Uppgift 3 *Terminering* (3p)
Förklara hur Dijkstras "token detection"-algoritm för att upptäcka terminering fungerar. Förklara algoritmen med hjälp av ett "illustrativt" exempel med 4 processorer.

Uppgift 4 *Principer för design av parallella algoritmer* (4p)
För att uppnå hög prestanda krävs inte bara snabba processorer utan även att data finns tillgängligt när operationerna ska utföras. För att uppnå hög prestanda i ett program måste man därför ta hänsyn till systemets minneshierarki.

Beskriv hur man vanligtvis försöker organisera beräkningarna (i samband med täta matriser) för att utnyttja minneshierarkin på bästa sätt.

Definiera begreppen temporal lokalitet och spatial lokalitet och hur dessa påverkas av olika typer av blockningstekniker (för beräkningar och datalagring).

Uppgift 5 *Glesa matriser* (3p)
Vad karakteriserar en gles matris? Beskriv 2 lagringssätt som används för att spara minne. Tala också om till vilken typ av matriser de passar.

Uppgift 6 (4p + 2p + 2p = 8p)
Givet en parallelldator med ett 2-dimensionellt kvadratiskt *store-and-forward* kommunikationsnätverk med *wrap-around*.

a. Konstruera en algoritm för att utföra en *k-to-all broadcast*, dvs en operation där k processorer samtidigt gör broadcast till alla andra. Antag att $k \leq \sqrt{p}$ och att de k sändande processorerna befinner sig i olika rader i processornätet.

b. Härled tiden för att utföra operationen i uppgift **a**. Hur förhåller sig denna tid i det generella fallet till tiden för att utföra en vanlig broadcast från endast en processor.

c. Vilka modifieringar behöver göras för att algoritmen ska klara av fall då flera sändande processorer befinner sig i samma processorrad? Vilken skillnad ger detta för tiden att utföra operationen?

Uppgift 7 *Delsekvenser* (4+2=6p)
Givet en sekvens $A =< a_1, a_2, \ldots, a_n >$ ges en delsekvens (*common-subsequence*) genom att ta bort några element ur A. (M.a.o. behöver delsekvensens element inte vara konsekutiva element från A, men de måste vara i samma ordning som i A.) Exempelvis är $< b, c, e >$ en delsekvens av $< a, b, c, d, e, f >$ medan $< b, c, g >$ och $< c, b, e >$ inte är det. Den längsta gemensamma delsekvensen (*longest-common-subsequence*) av två sekvenser $A =< a_1, a_2, \ldots, a_n >$ och $B =< b_1, b_2, \ldots, b_m >$ är den längsta sekvens som är delsekvens av både A och B.

Låt $F[i, j]$ vara längden av den längsta gemensamma delsekvensen av $< a_1, a_2, \ldots, a_i >$ och $< b_1, b_2, \ldots, b_j >$. Då är $F[n, m]$ längden av den längsta delsekvensen av A och B. $F[n, m]$ kan härledas från

$$F[i,j] = \begin{cases} 0 & \text{om } i = 0 \text{ eller } j = 0 \\ F[i-1, j-1] + 1 & \text{om } i, j > 0 \text{ och } a_i = b_j \\ \max\{F[i, j-1], F[i-1, j]\} & \text{om } i, j > 0 \text{ och } a_i \neq b_j \end{cases}$$

a. Konstruera en effektiv parallell algoritm för att beräkna $F[n, m]$ på ett två-dimensionellt nät med p processorer där $m, n > p$. (För att undvika krångel med specialfall kan du själv sätta och ange förutsättningar på relationer mellan p, m och n.)

b. Härled uttryck för totala exekveringstiden T_p, uppsnabbningen S och effektiviteten E för algoritmen. Är algoritmen kostnadsoptimal?