

Seismisk modellering på arbeidsstasjoner i nettverk

Fredrik Manne*

Norsk Hydro a.s Bergen, U&P Forskningscenter

1 Innledning

Vi har tatt en program kode for seismisk modellering skrevet i Fortran 77 og konvertert denne for kjøring på både arbeidsstasjoner i nett og på en dedikert parallell MIMD (Multiple Instructions, Multiple Data) datamaskin. Koden er en implementasjon av “Tip Wave Single Superposition” metoden. I denne rapporten beskriver vi arbeidet som ble lagt ned for å konvertere programmet samt de resultater vi har oppnådd.

Hoved-resultatet er at programmet er velegnet for parallell prosessering både på arbeidsstasjoner i nettverk og på en dedikert parallell datamaskin. Vi viser også at for denne applikasjonen er kjøretiden ved bruk av arbeidsstasjoner i nettverk konkurransedyktig med kjøretiden for en dedikert parallell datamaskin.

I seksjon 2 gir vi en grov beskrivelse av det aktuelle data programmet vi har sett på. Seksjon 3 beskriver tilgjengelige data ressurser ved Norsk Hydros forskningscenter i Bergen. I seksjon 4 beskriver vi de ulike parallelle arkitekturene som vi har benyttet. Utviklingen av det parallelle data programmet er omtalt i seksjon 5 og testene av det i seksjon 6, før vi oppsummerer i seksjon 7.

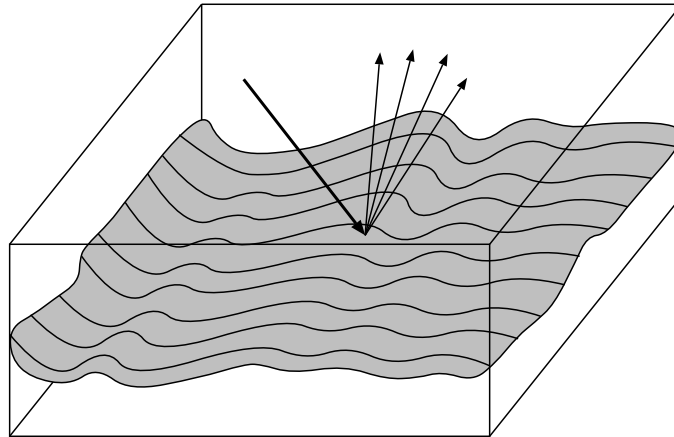
2 Super Posisjon

Super Posisjon er et program for seismisk modellering utviklet ved geoseksjonen ved Norsk Hydros forskningscenter i Bergen. Det baserer seg på teori utviklet av Klem, Musatov. et.al. [5] om spredning av bølger som treffer kanter og hjørner av en flate. Programmet brukes til å bedømme kvaliteten av en modell som beskriver en geologisk struktur. Modellen er utviklet på grunnlag av seismiske undersøkelser. Super Posisjon beregner effekten av å

*Seksjon for informatikk, Forskningscenteret, Norsk Hydro, N-5020 Bergen, fmanne@bg.nho.hydro.com.

foreta tilsvarende seismiske undersøkelser på den syntetiske modellen. Gjennom å sammenligne resultatet fra programmet med de virkelige data kan man vurdere hvor godt modellbeskrivelsen samsvarer med de virkelige forhold.

Simuleringen foregår gjennom at man avfyrer “skudd” som sender bølger ned i modellen. Man beregner så hvordan disse blir reflektert opp igjen til overflaten. For hvert skudd defineres et geografisk vindu som det er interessant å beregne effekten i. Beregningene gjøres ved hjelp av strålefølgning (figur 1).



Figur 1: Skudd sendes ned i modellen og reflekteres opp

Resultatet fra hvert skudd kan beregnes uavhengig av hverandre. I tillegg er også mange av beregningene for hver enkelt stråle uavhengige. Dersom størrelsen på hvert vindu er av størrelsesorden $O(n^2)$ så tar beregningene $O(n^3)$ tid (med en forholdsvis stor multiplikativ konstant). Resultatet fra hvert vindu blir fortløpende skrevet ut til en fil.

Da det er store datamengder som er involvert i beregningene er det behov for datamaskiner med både stor regnekapasitet og med stor hukommelse. Det er derfor allerede utviklet en versjon av Super Posisjon for kjøring på en SIMD (Single Instruction, Multiple Data) parallell datamaskin [7] og en versjon for kjøring på “Shared memory MIMD” maskiner [6]. Begge disse programmene utnytter parallelliteten innenfor hvert vindu. Programmet som beskrives her utnytter imidlertid parallelliteten mellom de ulike vinduene.

3 EDB ressurser ved Norsk Hydros Forsknings-senter

Ved Norsk Hydros forsknings-senter i Bergen finnes det idag i overkant av 50 Unix arbeidsstasjoner sammenkoblet i nett (Ethernet og FDDI). Disse er av en rekke ulike arkitekturer (F.eks. Sun, Sgi, Ibm, Dec, Alpha). I tillegg finnes det en rekke PCer også disse koblet i nett. Forsknings-senteret har dessuten

tilgang på dedikerte parallelle datamaskiner ved Institutt for informatikk ved Universitet i Bergen. Aktuelle maskiner her er MIMD-maskiner (Paragon) og SIMD-maskiner (MasPar).

4 Parallel Virtual Machine (PVM) og Paragon

Vi beskriver her de ulike parallelle datamaskiner vi har benyttet i våre forsøk. Først gir vi en oversikt over den programvaren vi har benyttet for å konfigurere flere arbeidsstasjoner som en felles regneressurs. Deretter omtaler vi den dedikerte parallelle datamaskinen vi har brukt.

PVM er “public-domain” programvare utviklet ved Oak Ridge National laboratories, USA [2, 3]. Det er et meldingsbasert system som gjør at en bruker selv kan konfigurere flere arbeidsstasjoner som en distribuert parallelle datamaskin koblet sammen vha. et nettverk (Ethernet, FDDI etc.). Dette gjør at man kan bruke allerede innkjøpte datamaskiner som en felles regneressurs.

Programutvikleren får tilgang på kommunikasjonsrutiner som kan brukes for å utveksle data mellom ulike prosesser. Eksempler er rutiner for sending og mottak av data. Når en datapakke har blitt sendt ut fra en prosess til en annen sørger PVM-systemet for at dataene blir levert til rett prosess på rett maskin og i rett format. Det er idag mulig å kjøre PVM på stort sett alle tilgjengelige typer arbeidsstasjoner (F.eks. Sun, Alpha, Sgi, Ibm).

Hver prosess som startes opp på en maskin under PVM, må konkurrere om maskinressurser med andre prosesser på denne maskinen. Det gjør at ytelsen til et distribuert program kan være følsomt for at flere brukere har tilgang til de samme maskiner.

Vi har hentet PVM versjon 3.2.6 over Internet og installert det på et utvalg maskiner ved Forskningscenteret. Sammen med PVM har vi også installert en del verktøy for å støtte program utvikling [1, 4]. Målinger vi har foretatt av den tilgjengelige båndbredden mellom ulike maskiner viser at den varierer mellom 0.5 Mbytes/sek. til 5 Mbytes/sek. Bestemmende faktorer her er hvilken type nett man bruker (Ethernet, FDDI), pakke størrelse samt hvilken type logisk oppkobling man har mellom de ulike maskinene.

Vi har gjennom Institutt for informatikk, UiB, hatt tilgang på en Intel Paragon dedikert parallell datamaskin. Dette er en lokal-hukommelse MIMD maskin utstyrt med 98 stk. i860 prosessorer samt ekstra prosessorer for å håndtere I/O. Maskinen fungerer på alle måter som arbeidsstasjoner sammenkoblet vha. PVM. Den vesentligste forskjellen ligger i at Paragon maskinen har et dedikert nettverk mellom prosessorene med meget høy båndbredde (31 Mbytes pr. sekund). Dessuten får brukeren eksklusiv tilgang til de prosessorene han har fått allokert. Det finnes også et disk array tilgjengelig, noe som muliggjør rask I/O. PVM finnes også installert på Paragon maskinen, men for å oppnå maksimal ytelse er det å foretrekke at man bruker et

lokalt kommunikasjonsbibliotek (NX). Dette har en noe lavere funksjonalitet en PVM, men er for alle praktiske formål identisk med PVM.

5 Parallele versjoner av Super Posisjon

Vi har skrevet om Super Posisjon programmet slik at det nå er et hoved program som leser inn alle data og distribuerer disse til en rekke slave program. Slave programmene tar imot dataene for et vindu om gangen, prosesserer dataene og sender de ferdige resultatene tilbake til hovedprogrammet som i sin tur skriver ut dataene til disk. Etter at prosesserte data fra et vindu er mottatt av hovedprogrammet sendes data for et nytt vindu ut.

Hovedprogrammet lagrer data fortløpende til disk. Dersom det er tilstrekkelig arbeidsoppgaver gjør dette at man får en god dynamisk lastbalansering mellom de ulike maskinene. Dersom en maskin går langsommere så vil den heller ikke få tildelt så mye arbeid som en maskin som går raskere. En følge av denne strategien er at resultatfilen vil være uordnet når programmet er ferdig. For å ordne dataene foretar vi derfor en postprosessering av dataene hvor de blir sortert og skrevet ut igjen til fil.

Hovedprogrammet for Paragon programmet tar imot dataene i samme rekkefølge som de ble sendt ut. Dette gjør at det ikke er behov for postprosessering av dataene. Dette kan føre til at et slave program må vente før hovedprogrammet er ferdig med å prosessere foregående vinduer, før det får tilsendt et nytt vindu. Dette er imidlertid ikke et stort problem på Paragon maskinen da man har tilgang på dedikerte homogene prosessorer. Det ble også vurdert å bruke det parallele disk arrayet på Paragon maskinen for å få til raskere I/O. Dette ble heller ikke aktuelt da I/O viste seg å ikke være en flaskehals.

Til kommunikasjon ble PVM benyttet på arbeidsstasjonene og NX på Paragon maskinen.

SIMD versjonen av Super Posisjon utfører bare en del av beregningene på hvert vindu. Dette skyldes at de siste beregningene på dataene var tenkt utført på en arbeidsstasjon. Det er imidlertid disse som er de mest arbeidskrevende. Det er derfor ikke meningsfullt å sammenligne tider fra denne implementasjonen med de som er beskrevet her.

Versjonen som ble laget for bruk på "Shared memory MIMD" maskiner utførte deler av strålefølgingen innenfor hvert vindu parallelt. Denne koden viste imidlertid liten forbedring ved bruk av flere CPUer.

6 Tester

Til kjøring av tester har vi brukt en samling heterogene arbeidsstasjoner lokalisert på Norsk Hydros forskningscenter samt Paragon maskinen ved

	Arbeidsstasjon	Relativ hastighet	Tid	Speed-up
1	Dec 3000 Mod. 500	1.0	38:03	1.0
2	Ibm RS6000 Mod. 990	0.97	21:12	1.79
3	Ibm RS6000 Mod. 580	0.69	14:08	2.69
4	Dec 3000 Mod. 500	0.59	12:04	3.15
5	Sparc 10 Mod. 512 med 2 pros.	0.49	11:06	3.43
6	Sparc 10 Mod. 512 med 2 pros.	0.49	10:17	3.70
7	IBM RS6000 Mod. 350	0.43	9:48	3.88
8	IBM RS6000 Mod. 320H	0.26	8:52	4.29
9	Iris Indigo R4000	0.21	8:04	4.72

Tabell 1: Tabellen viser arbeidsstasjoner som ble benyttet ordnet etter relativ hastighet. Den viser og kjøretid og speed-up for hver maskin som ble lagt til den virtuelle maskinen.

Institutt for informatikk, UiB. Tabell 1, kolonne 1, viser de ulike arbeidsstasjoner som er brukt ordnet etter relativ hastighet. Disse målingene er gjort ved å kjøre et slaveprogram på hver maskin (to på Sparc 10) og hovedprogrammet på en Sparc 10. De reflekterer kun maskinens hastighet slik den var tilgjengelig til oss. Vi hadde ikke eksklusiv tilgang på maskinene, men testene ble utført når belastningen var lav. Merk også at tilgjengelig hukommelse kan påvirke regnehastigheten. Den eneste optimaliseringen av koden som vi har foretatt er å kompilere koden med høyest tilgjengelig optimaliseringsnivå.

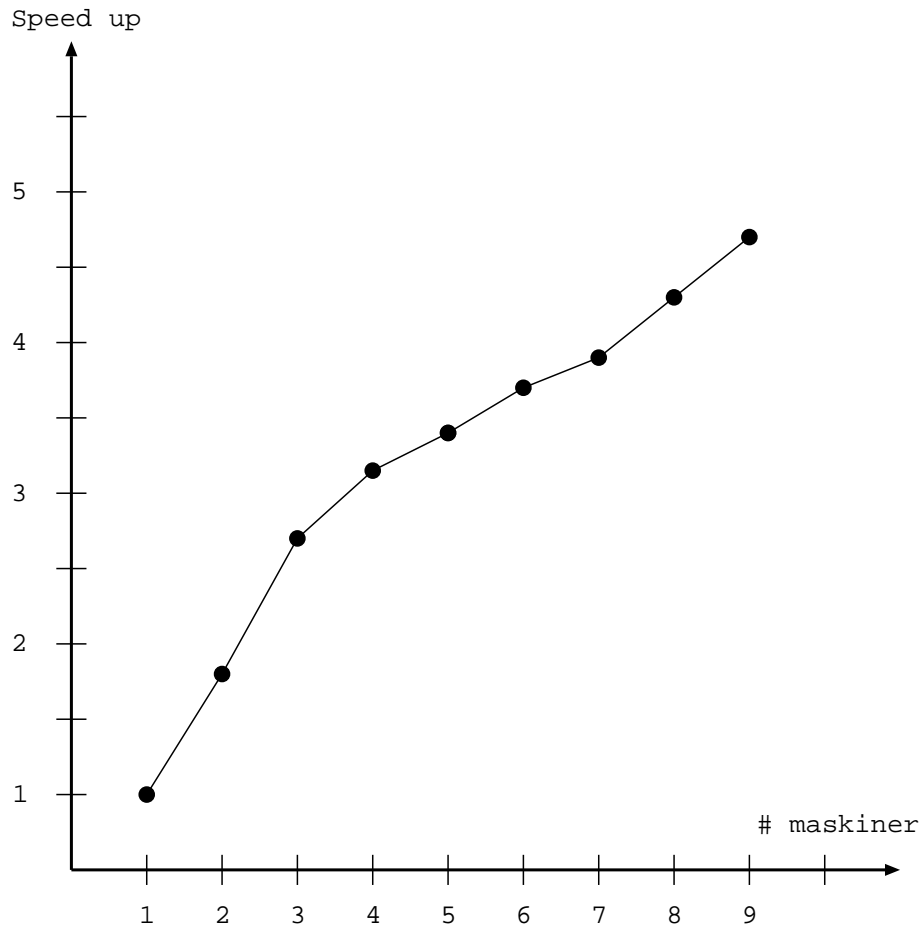
Ved utføring av programmet ble hovedprogrammet lagt til en Sparc 10. Dvs. at denne maskinen i tillegg til to slaveprogram også betjente hovedprogrammet i samtlige konfigurasjoner.

Data settet som ble brukt besto av 263 skudd. Gjennomsnittlig data-mengde som ble sendt ut var 209 Kbytes pr. skudd og 29 Kbytes data ble mottatt for hvert vindu etter ferdig prosessering.

I tabell 1, kolonne 3, viser vi kjøretid (minutter:sekunder) og i kolonne 4 speed-up målt ut ifra kjøretiden på den raskeste maskinen. For hver linje i tabellen er et nytt slaveprogram lagt ut på den angitte maskinen, i tillegg til at et slaveprogram ble kjørt for hver av de ovenstående linjene i tabellen. I figur 2 er samme resultat fremstilt grafisk. Speed-up er målt som den tiden det tar å utføre det beste sekvensielle programmet på den raskeste maskinen delt på den tiden det tok å utføre programmet på den virtuelle parallelle maskinen.

Som det fremgår av tallene i tabell 1 følger speed-up til programmet tilnærmet lineært med regnekraften til maskinene. På en Iris Indigo som programmet opprinnelig ble skrevet for, var kjøretiden 218.5 minutter og speed-up er en faktor 27.

Sortering av data tar på en Sparc 10 ca. ett minutt. Det er trolig at



Figur 2: Speed-up i forhold til den kraftigste arbeidsstasjonen

denne tiden kan reduseres dersom man hadde gjort mere bruk av uformatert utskrift. Programmet som tolker data fra Super-posisjons programmet er imidlertid avhengig av et bestemt filformat.

Resultatene fra kjøring av Super Posisjon på Paragon er fremstilt i tabell 2 og i figur 3. Antall prosessorer som ble brukt er en større enn antall slaveprogram. Dette skyldes at hovedprogrammet kjører på en egen prosessor. Som det fremgår av tabellene tar det omtrent like lang tid å kjøre på en full konfigurasjon av arbeidsstasjoner som det tar å kjøre på 45 noder av Paragon maskinen. Også på Paragon er forbedringen av kjøretiden tilnærmet lineær.

7 Konklusjon

Vi har utviklet en MIMD versjon av et program for seismisk modellering. Dette har vi kjørt både på arbeidsstasjoner i nettverk og på en dedikert parallell datamaskin. Resultatet er at det er mulig å kjøre programmet raskere og med tilgang på mer hukommelse. Vi har vist at enkle lastbalanseringsalgo-

# Slave pros.	Tid	Speed-up
1	289:14	1.0
5	58:53	4.90
10	30:43	9.32
15	20:35	14.10
20	16:14	18.06
25	12:59	22.23
30	10:58	26.27
35	9:39	28.90
40	9:21	32.11
45	8:12	36.13

Tabell 2: Kjøretid på Paragon

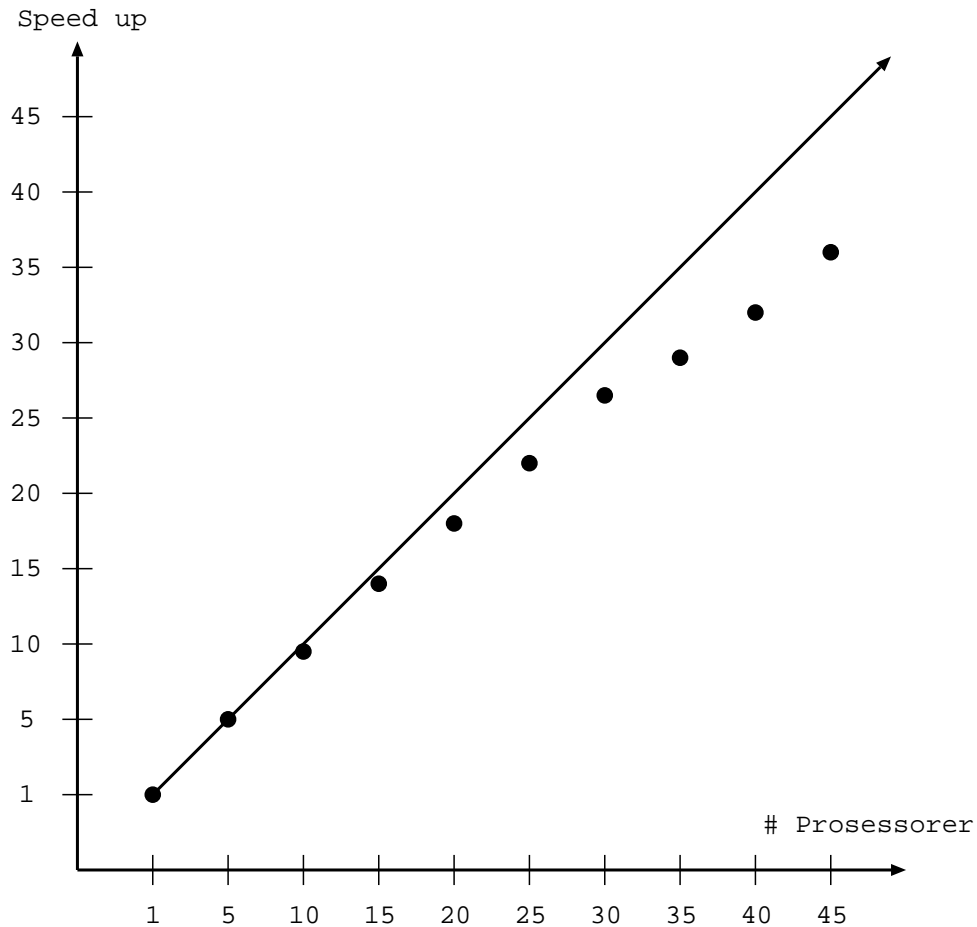
ritmer gir programmet tilnærmet lineær speed-up både ved kjøring på arbeidsstasjoner i nett og på en Paragon maskin. PVM versjonen av programmet var ikke begrenset av den lavere båndbredden for kommunikasjon.

Det var forholdsvis opplagt hvordan programmet skulle skrives om. Strukturen til programmet med separate “vinduer” gav en naturlig oppdeling av jobbene som skulle utføres. Arbeidet som har blitt lagt ned har gått med til å omstrukturere programmet og få kommunikasjonen mellom de ulike prosessorene til å fungere. Den totale arbeidsmengden har vært cirka et månedsverk.

Det er trolig at man både på arbeidsstasjonene og på Paragon maskinen kunne ha oppnådd bedre ytelse dersom man hadde fin-tunet programmet mere og eventuelt brukt ferdig optimaliserte biblioteker. Dette ble ikke gjort da det ble ansett å ligge utenfor rammen av dette prosjektet. En slik fin-tuning ville dessuten ha medført økte kostnader.

En mulighet for å videreutvikle programmet er å bygge inn kapasitet for feiltoleranse. Dette er særlig aktuelt dersom programmet blir benyttet i et produksjons miljø. Dersom en eller flere av arbeidsstasjonene som utfører slaveprogram går ned vil programmet ikke kunne fullføre sin beregning. For å sikre seg mot dette kan man bruke PVM rutiner som registrerer status til de ulike prosessene som inngår i den virtuelle maskinen.

Det er trolig at flere dataprogrammer utviklet for petroleumsvirksomhet har en tilsvarende oppbygging som Super Posisjon, med fullstendig eller nesten fullstendig uavhengige beregninger som skal foretaes over et geografisk område. Slike dataprogram vil med stor sannsynlighet også kunne modifiseres forholdsvis lett for kjøring på ulike distribuerte plattformer. I og med at den eneste kostnaden er selve omskrivingen av dataprogrammet ligger det her et stort potensiale for å oppnå økt ytelse uten store investeringer.



Figur 3: Speed-up på Paragon

References

- [1] A. L. BEGUELIN, *Xab: A tool for monitoring PVM programs*, in Proceedings Workshop on Heterogeneous Processing WHP'93, IEEE Computer Society Press, 1993, pp. 92–97.
- [2] J. DONGARRA, G. A. GEIST, R. MANCHEK, AND V. S. SUNDARAM, *Integrated PVM framework supports heterogeneous network computing*, Computers in Physics, 7 (1993), pp. 166–175.
- [3] G. A. GEIST, A. BEGUELIN, J. DONGARRA, W. JIANG, R. MANCHEK, AND V. S. SUNDARAM, *PVM 3 user's guide and reference manual*, Oak Ridge National Laboratory, 1993.
- [4] M. HEATH AND J. ETHERIDGE, *Visualizing the performance of parallel programs*, IEEE software, 8 (1991), pp. 29–39.

- [5] K. D. KLEM-MUSATOV, A. AIZENBERG, AND G. A. KLEM-MUSATOVA, *An algorithm for mathematical modelling of three dimensional diffraction fields*, Soviet geol. and Geophys, 23 (1982), pp. 116–121.
- [6] F. MANNE, *A test of the SGI challenge parallel computer using software from Norsk Hydro*. Norsk Hydro, internal report, 1993.
- [7] T. SØREVIK AND S. D. SINGH, *Parallelizing a Fortran-77 code for seismic simulation on the MasPar MP-1*, Tech. Report CS-93-77, Department of Informatics, University of Bergen, 1993.