DINAMIČNO DODELJEVANJE PROCESOV NA POLJU TRANSPUTERJEV

INFORMATICA 4/91

Keywords: parallel processor system, token data flow, transputer, topology, simulation, dynamic scheduling

Peter Zaveršek, VELCOM, Velenje
Peter Kolbezen, Institut Jožef Stefan, Ljubljana

V članku je obravnavana problematika dodeljevanja procesov na večprocesorskem vezju, ki ga sestavljajo večkratno povezane zanke. Lokacija procesiranja posameznega procesa ni vnaprej določena. Obravnavani so trije načini dodeljevanja procesov: pri enonivojskem načinu prenosa podatkov po eni zanki ter pri enonivojskem in dvonivojskem načinu prenosa podatkov po dveh zankah hkrati. Za vse tri primere dinamičnega dodeljevanja je izdelan simulator dodeljevanja procesov. S pomočjo simulatorja sta analizirani izkoriščenost polja in hitrost izvajanja algoritma v odvisnosti od parametrov, kot so velikost in konfiguracija procesorskega polja, topologija programskega grafa in časova karakteristika procesov. Na osnovi opravljene analize je predlagan najboljši način dodeljevanja procesov in za različne topologije programskega grafa takšen izbor parametrov, ki vodi k čim večji učinkovitosti večprocesorskega sistema.

PROCESS ALLOCATION IN THE MULTITRANSPUTER NETWORK. The paper deals with a problem of process allocation in a multitransputer network which consists of multiple-connected unidirectional rings. Executing location of every certain process is not explicitly stated, i.e. process allocation is dynamic. Three different possible algorithms for process allocation are presented. The algorithms are as follows: a) one-level communication and sending data in one direction, and b) one-level communication and sending data in two directions simultaneously, c)two-level communication and sending data in two directions simultaneously. Efficiency of different network configurations, program graph topologies and ratio of process execution time to process transfer time were verified by a simulation. Finally the results of simulation are presented and the most efficient process allocation algorithm of the three proposed is selected.

1. UVOD

Zahtevam po vedno večji moči procesiranja je mogoče zadostiti le s paralelnim računanjem na dovolj veliki množici procesorjev. Takšno množico, ki ima paralelno strukturo, lahko predstavlja polje procesorjev. Posameznih procesorjev polja ni mogoče izkoristiti tako dobro, kot pri enem samem procesorju. Procesorji, ki delujejo paralelno, praviloma niso zasedeni ves čas izvajanja algoritma. Izkoriščenost (zasedenost) po procesorjih je neenakomerno porazdeljena. Idealnim razmeram se je mogoče le bolj ali manj približati. Pri tem igra pomembno vlogo več dejavnikov. Pomembnejši med njimi so:

Paralelizacija algoritma. Postopek naj bi odkril ves inherentni paralelizem v danem algoritmu, ki se bo izvajal na paralelnem sistemu. Z ustrezno analizo algoritma ugotovimo, kateri procesi se lahko izvajajo paralelno in kateri sekvenčno, čas izvajanja posameznih procesov in število podatkov, ki so za izvajanje potrebni.

Granulacija algoritma. Algoritem, ki določa neko opravilo in ga izvajamo na našem sistemu, razstavimo na več podopravil. Na koliko podopravil razbijemo opravilo glede na velikost celotnega opravila, kar določa razdrobljenost (granulacijo) algoritma, je odvisno od števila in od zmogljivosti razpoložljivih procesorjev. Podopravilo (subtask ali grain) se avtonomno izvaja na enem od procesorjev polja. Granulacija vpliva na razmerje med časom, ki je potreben za prenos podatkov med dvema procesorjema in časom izvajanja procesa. To razmerje imenujemo količnik prenosnega časa. Izbira najustreznejše granulacije pri danem večprocesorskem sistemu je odvisna od topologije in velikosti procesorskega sistema in ima velik vpliv na učinkovitost izvajanja algoritma. Zato moramo pri pripravi algoritma za izvajanje na paralelnem sistemu granulaciji posvetiti še posebno pozornost.

Dodeljevanje procesov. Procese moramo razporediti po polju procesorjev tako, da je polje čimbolj izkoriščeno. Pri tem je pomembno, da je število komunikacij med procesi čimmanjše in čas dodeljevanja čimkrajši.

Porazdelitev bremena. Procesorji v polju morajo biti enakomerno obremenjeni. Neenakomerna obremenjenost zmanjša učinkovitost polja in poveča ceno, ki je posledica povečanega skupnega časa procesiranja.

V delu je obravnavana problematika, ki je prisotna predvsem v zadnjih dveh dejavnikih, in se opira na simulacijo predlaganega večprocesorskega sistema. Sistem je sestavljen iz polja transputerjev in je podrobneje opisan v delu /6/. Zanj so značilni:

Krožna konfiguracija: Konfiguracijo sistema sestavlja večje število med seboj prepletenih prstanov (linkov), ki povezujejo vse transputerje polja. Oblika poti ni pomembna. Pot mora biti le krožno zaključena neglede na to, kje v sistemu ima izhodiščno točko. Primer takšnega sistema je hiperkocka.

Hierarhično dodeljevanje procesorjev: Polje procesorjev izvaja podopravila. V podopravilu se operacije (imenovane procesi) izvajajo, kolikor je mogoče konkurenčno. Procesi so v značilnem odnosu, toda neodvisni med seboj (tako kot DO zanke v fortranu). Podopravila je mogoče predstaviti v obliki hierarhičnih acikličnih grafov pretoka podatkov (HADFG). Vozlišče takšnega podgrafa je proces, ki se izvaja na enem od transputerjev polja. Znotraj vsakega procesa so možne zanke. Kontrolo nad podopravili, ki so določeni z eno ali več HADFG, ima poseben procesor na vhodu v procesorsko polje.

Dinamično dodeljevanje procesorjev: Razvrščanje procesov je avtomatizirano in dinamično. Dosledno sledi pravilu, ki ga določa položaj procesa ali veje v strukturi HADFG.

Komunikacija s pomočjo žetonov: Komunikacijski mehanizmi so osnovani na uporabi žetonov, ki krožno potujejo znotraj posameznih prstanov. Komunikacije so enosmerne. Fizični prstan sestavljata dva komunikacijska prstana. Eden od njiju je uporabljen za prenos krmilnih žetonov, drugi pa za prenos podatkov, programskih blokov in rezultatov.

2. DODELJEVANJE PROCESOV

Poznanih je več načinov dodeljevanja procesov. Bistveno se med seboj razlikujejo predvsem načini statičnega in dinamičnega razporejanja procesov.

Pri statičnem razporejanju preslikamo posamezne procese na določene procesorje že pred začetkom izvajanja. Preslikava je osnovana na analizi sočasnosti in čimboljši izkoriščenosti procesorjev. Praviloma je pri statičnem razporejanju mogoče doseči večjo izkoriščenost sistema.

Pri dinamičnem razporejanju se preslikajo posamezni procesi na procesorje med samim izvajanjem algoritma. Izkoriščenost procesorjev, ki mora biti v vsakem trenutku čimboljša, se dinamično spreminja in je odvisna od vsakokratnega trenutnega stanja sistema.

Uporaba dinamičnega načina razporejanja je posebej primerna v naslednjih primerih:

- čas izvajanja procesov ni vnaprej določen,
- v fazi analize algoritma ne poznamo konfiguracije sistema, na katerem se bo algoritem izvajal,
- želimo trdoživ (fault tolerant) sistem.

2.1. DINAMIČNO RAZPOREJANJE PROCESOV

Sistemi z dinamičnim dodeljevanjem procesov ugotavljajo trenutno stanje sistema na več načinov /7/:

- **Bolj obremenjen procesor išče manj obremenjenega.** Ta način je uspešnejši za malo in srednje obremenjene procesorje.
- **Manj obremenjen procesor išče bolj obremenjenega.** Rešitev je bolj primerna za zelo obremenjene procesorje.

Naš sistem uporablja oba zgoraj omenjena načina. Obremenjeni procesor išče nezaseden procesor, da bi mu predal enega izmed čakajočih procesov. Ko ga najde, mu preko sporočila preda podatke o procesu, ki je prvi na vrsti na čakajoči lestvici procesov. Postopek se ponavlja, dokler procesor ne ugotovi, da ni več nezasedenega procesorja. Takrat se postavi v stanje čakanja. Procesor, ki zaključi svoj proces in se tako sprosti, pošje procesorjem v zanki sporočilo, da ni več zaseden. Tedaj mu pošlje procesor, ki ima še čakajoče procese, zahteve oz. podatke o naslednjem procesu, ki ga bo izvajal in cikel se tako ponovi /6/.

2.2. TEHNIKA ZASEGANJA VIROV IN ZASEDENOST SISTEMA

Poznanih je več tehnik zaseganja sistemskih virov. Zlasti sistemi hierarhične zgradbe imajo lahko poseben nadzorni procesor, ki hrani informacijo o stanju vseh ostalih (delovnih) procesorjev. Nadzor nad zasedenostjo procesorjev je osredotočen na enem samem mestu. Slabost takega načina so dodatne komunikacije, ki so potrebne zaradi nenehnega testiranja stanj sistema. Ker so podatki o sistemu shranjeni v skupni tabeli, do katere imajo dostop vsi procesorji, obstaja možnost sočasnega dostopa in

lahko pride do konfliktne situacije. V primeru, da ima vsak procesor svojo kopijo tabele, ki vsebuje stanja vseh procesorjev sistema, je nadzor nad zasedenostjo procesorjev enako porazdeljen na vse procesorje. Pri tem je potrebno vse kopije sproti osveževati. Možnost konfliktnih situacij je v tem primeru lahko ob primerni organiziranosti sistema manjša, z večanjem števila procesorjev v sistemu pa se hitro veča tudi število komunikacij.

V obravnavanem sistemu je izdelana posebna tehnika zaseganja sistemskih virov, pri kateri je nadzor nad zasedenostjo procesorjev porazdeljen le nekaterim procesorjem, t.im. korenskim procesorjem v sistemu. Ti imajo evidenco o zasedenosti tistih procesorjev, ki so korenskemu procesorju dosegljivi.

2.3. PRENAŠANJE SPOROČIL

Način prenašanja sporočil ima pomembno vlogo v večprocesorskem sistemu. Sporočilo je sestavljeno iz niza elementarnih podatkovnih paketov FCD (imenovanih "flits" - Flow Control Digits). FCD je najmanjša enota sporočila, ki jo lahko kanal ali vrsta sprejme ali zavrne. Večja hitrost prenosa preko večih posrednikov (procesorjev v sistemu) je dosežena tako, da vsak posrednik prične predajati sporočilo naslednjemu procesorju, še predno ga sprejme v celoti. Predaja prvega FCD se začne takoj, ko procesor preveri, da sporočilo ni namenjeno njemu. Posamezne FCD sporočila pa lahko začasno tudi zadrži, če je pot naprej zasedena. Takšno prenašanje sporočila, ki leze po sistemu kot črv, je v tuji literaturi poznano pod imenom "cut-through" /8/.

Pri prenosu sporočila ne sme nastopiti smrtni objem (deadlock) ali živi objem (livelock). Smrtnemu in živemu objemu se v našem transputerskem sistemu izognemo že na dokaj enostaven način. Smrtni objem preprečimo z dovolj velikimi vmesniki za prenos sporočil pri prenosnih procesorjih /2,8/, živi objem pa z enosmernimi zankami.

3. MEDTRANSPUTERSKE KOMUNIKACIJE

Transputer uvrščamo v družino RISC procesorjev /5/. Na tržišču je že več transputerjev, ki se med seboj razlikujejo predvsem po svoji zmogljivosti (hitrosti), protokolu komuniciranja, podpori za aritmetične operacije in pd., lahko pa tudi po namembnosti, ki jo določi že proizvajalec. Za nas je pomembna značilnost, da transputer že na strojnem nivoju podpira paralelno izvajanje procesov (v določeni meri kvaziparalelno). Komunikaciji so namenjene štiri vgrajene dvosmerne zanke (linki). Te zanke omogočajo, da lako transputerje medsebojno povezujemo brez dodatnih vmesnikov. Komunikacija preko zanke lahko v obeh smereh poteka istočasno, avtomatično in skoraj brez obremenjevanja transputerja. Uporabniku je zunanja komunikacija povsem identična komunikaciji med notranjimi procesi.

Prenos podatkov med transputerji je paketni in poteka preko komunikacijske zanke s hitrostjo, ki je po tovarniških podatkih 5, 10 ali 20 Mb/s. Tej vrednosti se lahko približamo, če imamo prenos v eni zanki in le v eni smeri, in če transputer ne opravlja nobenih drugih opravil. Realno dosežena hitrost prenosa je v veliki meri odvisna tudi od vrste uporabljenega transputerja.

Prenosni paket sestavlja 11 bitov:

1 bit	startni bit
1 bit	zastavica "podatki"
8 bitov	podatek
1 bit	stop bit

Transputer, ki sporočilo sprejema, mora potrditi vsako besedo sporočila posebej. Potrditveno sporočilo je dolgo 2 bita:

1 bit	startni bit
1 bit	stop bit

Transputer T414 /4/ prenaša sporočilo tako, da najprej sprejme celo besedo sporočila (11 bitov) in nato odda potrdilo o sprejemu (2 bita). Teoretična hitrost prenosa je:

$$CT_{\text{T414}}^{\text{uni}} = \frac{20Mbs^{-1}}{13ciklov} = 1{,}54Mbs^{-1}$$

Transputer T800 ima izboljšan protokol prenosa. Ne čaka na sprejem cele besede sporočila, temveč potrdi sprejem sporočila že pred zaključkom prenosa. Oddana beseda in potrditev sprejema se med seboj prekrivata. Zato je za prenos sporočila potrebno manjše število ciklov. Dolžina sporočila je le 11 bitov:

$$CT_{\text{T800}}^{\text{uni}} = \frac{20Mbs^{-1}}{11ciklov} = 1{,}82Mbs^{-1}$$

Dvosmerni prenos preko ene zanke zahteva dodatne potrditvene bite za nasprotno smer, kar pomeni za T800 13 bitov in prenos s hitrostjo $CT = 1{,}54MBs^{-1}$. To je teoretična hitrost prenosa za vsako zanko (link) v vsaki smeri. Teoretična skupna možna hitrost

prenosa preko štirih dvosmernih zank je $12MBs^{-1}$, v eni smeri pa je potem okoli $1,5MBs^{-1}$.

Realna slika je seveda drugačna. Čeprav delujejo komunikacijske zanke ločeno in so ločene tudi od ostalih procesov, ki se izvajajo na transputerju, prihaja do medsebojnih vplivov. Realne vrednosti so zato dosti manjše od teoretičnih.

Prenosi preko zank (sprejem, oddaja, vmesno shranjevanje sporočil) zahtevajo dostop do transputerjevega pomnilnika. Prenosi do pomnilnika potekajo v obliki DMA prenosov in so načelno med seboj neodvisni, vendar zasedejo določen del pasovne širine transputerjevega podatkovnega in naslovnega vodila. Pasovna širina podatkovnega vodila znaša za notranji pomnilnik $80MBs^{-1}$, za zunanji pomnilnik pa $26.6MBs^{-1}$. Primerjava s teoretično najvišjo možno hitrostjo prenosa nam pokaže, da to pomeni 15% pasovne širine vodila notranjega pomnilnika oz. skoraj 50% pasovne širine vodila zunanjega pomnilnika. Posledice so dvojne. Konfliktne situacije po eni strani zmanjšujejo hitrost prenosov po zankah, po drugi strani pa ovirajo izvajanje samega procesa na transputerju, ki prav tako potrebuje dostop do pomnilnika. Realno so prenosi po zankah redkejši in je zato pogostost konfliktnih situacij manjša.

Procesi, ki izvajajo servisne rutine za prenose, in preklopi med delovnimi procesi (konteksti) nimajo zanemarljivega vpliva na učinkovitost procesiranja. Servisnih procesov je lahko več hkrati in običajno tečejo v prioritetnem načinu (PRI PAR, PRI ALT). Pogostost takšnih procesov lahko občutno podaljša čas izvajanja delovnega procesa in s tem delovno zmogljivost procesorja.

Servisne rutine za prenos podatkov morajo biti čimbolj učinkovite. Temu pripomore predvsem hitrost izvajanja teh rutin. Tudi promet podatkov naj bo redkejši, da je obremenjenost notranjih vodil manjša.

Analiza prenosa podatkov med transputerji in raziskava vpliva prenosa podatkov na učinkovitost sistema, sta opisana v delu /1/. Raziskava je bila omejena na transputer T414 pri taktu 15MHz in pri različnih konfiguracijah transputerskih povezav (prstan, drevesna struktura, hiperkocka). Za vsako konfiguracijo je bila napisana posebna servisna komunikacijska rutina. Testiranje je bilo opravljeno pri različnih dolžinah sporočil v razponu od enega do 1024 bytov.

Rezultati kažejo realen vpliv naštetih faktorjev na zmanjšanje delovne moči transputerja. Enosmerni prenos preko ene same zanke zmanjša moč procesiranja za 5 do 10%. Zanimivo je, da se minimalna moč ne pojavlja na gornji ali spodnji meji dolžine sporočila, temveč pri 64 bytih. Možna razlaga tega pojava je, da prihaja do časovno izredno neugodnega preklapljanja kontekstov.

Istočasni prenos sporočil po različnih zankah in tudi takih, ki dopuščajo dvosmerne prenose, dajejo približno enake rezultate. To velja le za prenose daljših podatkovnih blokov (okoli 1kB). Zmanjšanje delovne moči pri takšnih prenosih je okoli 12%. Krajša sporočila zahtevajo pogostejše preklope na servisni proces in delovna moč transputerja občutno pade - tudi za 85% in več. Zato kratka sporočila niso zaželena.

4. SIMULACIJA PROCESIRANJA

Predpostavljmo, da so podalgoritmi, ki se izvajajo na našem transputerskem polju večprocesorskega sistema, predstavljivi v obliki hierarhičnih acikličnih grafov pretoka podatkov (HADFG). To so drevesni grafi, katerih veje so razvrščene v več nivojev. Vsaka veja se končuje z novim razvejiščem ali s procesom. Nivoji z množicami paralelnih procesov in nivoji z množicami zaporednih procesov se med seboj izmenično prepletajo. Iz dela /6/ je razvidno, da taka predstavitev omogoča avtomatizirano dinamično porazdeljevanje procesov algoritma na polje procesorjev.

Dodeljevanje procesov smo opazovali na simulatorju, ki smo ga posebej zgradili v ta namen z nalogo, da čimbolj verno simulira izvajanje algoritma na našem procesorskem polju. Simulator upošteva zmanjšano delovno moč procesorja, ki je posledica prenosov. Predpostavljamo, da prenos v eni sami smeri zmanjša delovno moč transputerja na 90%, prenos v dveh smereh pa na 85%. Ta ocena je slabša od podane v literaturi /1/.

Simulacija dodeljevanja in izvajanja procesov naj bi dala odgovore na nekatera odprta vprašanja, ki so bila zastavljena že v delu /6/. Predvsem sta nas zanimala:

- najustreznejša granulacija algoritma (količnik prenosnega časa), in
- najustreznejša topologija (prstan, kvadrat, pravokotnik).

4.1. KONFLIKTNE SITUACIJE

V realnem večprocesorskem sistemu nastopajo konfliktne situacije, ki so pogojene z zgodovino izvajanja opravila. Simulator, ki bi simuliral delovanje transputerskega polja do vseh potankosti, bi imel veliko časovno in programsko kompleksnost. Ker smo želeli kompleksnost simulacije zmanjšati, smo jo poenostavili s hevrističnim razreševanjem konfliktov.

Če se pri simulaciji pojavi konfliktna situacija, izberemo enega izmed udeležencev in mu damo prednost pred ostalimi. Posledica tako poenostavljenega razreševanja konfliktov je, da nam lahko dajejo ponavljajoče se simulacije istega HADFG na istem sistemu različne rezultate. Zato smo predpostavili, da predstavlja pravilen rezultat povprečje rezultatov večjega števila simulacij.

4.2. NAČINI RAZPOREJANJA PROCESOV

Uporabili smo več različnih načinov razporejanja procesov. Vsi trije načini so osnovani na enakem osnovnem principu, ki je podrobneje opisan v delu /6/, in se opira na hierarhično drevesno strukturo grafa.

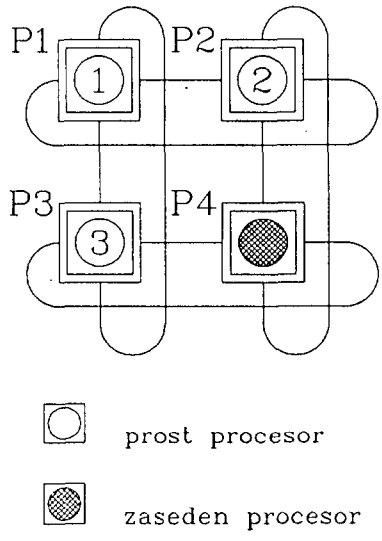
Razporejanje procesov se lahko izvaja v največ dveh nivojih. Procesorji v obeh zankah, ki pripadata začetnemu korenskemu procesorju tvorijo t.i. prvi procesorski nivo. Če je iskanje prostih procesorjev na tem nivoju neuspešno, nadaljujemo iskanje prostih procesorjev na drugem nivoju, na katerem pregledamo še preostali del ravninskega polja. Pregledovanje poteka tako, da pošlje korenski procesor procesorjem prvega nivoja v eni izmed dveh možnih zank (t.im. horizontalni ali vertikalni) zahtevo, da v zankah, ki so pravokotne na prvi nivo najdejo prost procesor. Oglejmo si posamezne načine razporejanja procesov:

a) Enonivojski način razporejanja s prenosom podatkov po eni zanki. Ta način je najpreprostejši. Osnovno idejo razporejanja najlažje predočimo s pomočjo paralelnega razvejišča grafa. Ko se spustimo po grafu do paralelnega razvejišča, zasedemo največ toliko prostih procesorjev, kolikor je vej v razvejišču. Dostopni procesorji se nahajajo neposredno v eni izmed dveh zank prvega procesorskega nivoja. Če pri zaseganju procesorjev zmanjka prostih procesorjev v prvi zanki, jih zasežemo še v drugi. Pri tem se lahko primeri, da je število procesov večje kot število dostopnih procesorjev. V takšnem primeru postavimo čakajoče procese v čakalne vrste. Ko se kateri izmed dostopnih procesorjev sprosti, mu dodelimo naslednji čakajoči proces.

Opisana metoda ima vsaj dve slabosti:

- Dodeljevanje procesov poteka v vsakem trenutku le v eni zanki (horizontalni ali vertikalni), čeprav lahko transputer hkrati predaja sporočila v več smereh.
- Ko pridemo v novo razvejišče, v fazi "priprave" takoj zasežemo vse razpoložljive procesorje. Zaseženi procesorji nadalje čakajo na sprejem podatkov o procesu, ki ga morajo izvajati. V stanju čakanja so blokirani in neuporabni, kar zmanjšuje izkoriščenost procesorskega polja.

b) Enonivojski način razporejanja s prenosom podatkov po dveh zankah hkrati. Posebnost tega načina je, da pri razporejanju ne rezerviramo vnaprej vseh dostopnih procesorjev. Izberemo le en prost procesor v vodoravni in le en prost procesor v navpični zanki. Prenos procesov lahko poteka hkrati v obeh smereh. Takoj, ko se eno od sporočil prenese, poiščemo naslednji prost procesor v isti smeri, tj. v smeri že prenešenega sporočila. Tako se lahko tudi faza "priprave" v eni smeri in faza prenosa sporočila v drugi smeri izvajata hkrati. To se dogaja v primerih, ko sta dolžini sporočil različni. V primerjavi z zgornjim načinom razporejanja so lastnosti tega načina naslednje:



Slika 1. Neizkoriščenost polja pri enonivojskem dodeljevanju

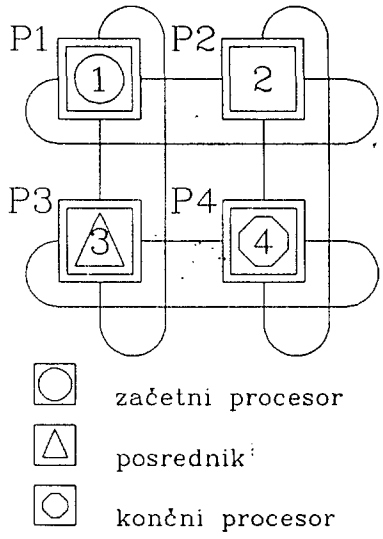
- Sistem je hitrejši zaradi istočasnega prenašanja sporočil v obeh smereh.
- Blokiranost procesorjev v polju je manjša.
- Delovna moč korenskega procesorja je zaradi povečanega obsega strežbe manjša.

Skupna slabost obeh gornjih načinov razporejanja je, da lahko paralelne procese dodeljujeta le v vodoravni in navpični zanki glede na korenski procesor, tj. le na prvem procesorskem nivoju. Večjega števila procesov, kot je prostih procesorjev v teh dveh zankah, korenski procesor ne more oddati. Zato obstaja možnost, da ostanejo nekateri procesorji polja neizkoriščeni.

Slika 1 kaže neučinkovitost enonivojskega razporejanja na izkoriščenost sistema. Na polju štirih transputerjev se lahko izvajajo štirje procesi hkrati, v našem primeru pa le trije. Četrti proces čaka na

procesor toliko časa, dokler se ne sprosti eden od treh že zasedenih procesorjev, medtem ko četrti procesor ostane ves čas neizkoriščen.

c) Dvonivojski način razporejanja pri prenosu podatkov po dveh zankah hkrati. Ta način omogoča, da so lahko hkrati izkoriščeni vsi procesorji polja. Za razliko od gornjih dveh metod iščemo tu nezasedene procesoje po celotnem procesorskem polju in ne le



Slika 2. Izkoriščenost polja pri dvonivojskem dodeljevanju

po zankah, ki potekata skozi začetni korenski procesor. Na prvem nivoju iskanja prostih procesorjev je način delovanja enak, kot pri prvih dveh načinih. Če je iskanje na tem nivoju neuspešno, ga nadaljujemo na drugem nivoju in tako pregledamo celotno ravninsko polje procesorjev. Iskanje prostih procesorjev na drugem nivoju poteka takole:

Začetni (korenski) procesor najpreje ugotovi zanko v katero ne oddaja nobenega procesa. Procesorji v izbrani zanki so procesorji prvega nivoja. Tem procesorjem pošlje žeton z zahtevo, da mu najdejo prost procesor na drugem nivoju. Procesorji, ki so v zanki, iščejo prost procesor drug za drugim. Procesor, ki je našel prost procesor, postane posrednik pri prenosu podatkov in sporočil.

Slika 2 kaže pri uporabi tretjega načina razdeljevanja procesov na danem primeru popolno izkoriščenost procesorskega polja. Ta način se je izkazal kot najboljši.

4.3. SIMULACIJA IZVAJANJA HADFG

Najpreje smo generirali množico G_n 20 naključnih grafov HADFG. Vseh 20 grafov smo obravnavali kot sortirane in nesortirane grafe. Vsak graf iz množice G_n smo tudi sortirali in tako dobili množico sortiranih grafov G_s. Skupaj smo simulirali izvajanje 2 x 20 grafov iz mnžice G, $\left(G = G_n \cup G_s\right)$. Pomen sortiranja grafov in definicijo sortiranega grafa najdemo v delu /6/.

Izvajanje vsakega grafa smo simulirali na različnih konfiguracijah procesorskega polja s številom procesorjev od 1 do 16. Izbirali smo konfiguracije iz nabora konfiguracij $K = a \times b$, kjer sta $a,b \in \{1,2,3,4\}$.

V prvi fazi raziskave smo opazovali čas izvajanja grafa v odvisnosti od izbranih konfiguracij. Na osnovi dobljenih rezultatov smo se pri nadaljnih raziskavah omejili na tiste konfiguracije polja, pri katerih so se pokazali boljši rezultati.

V drugi fazi raziskave smo opazovali čas izvajanja grafa HADFG v odvisnosti od količnika prenosnega časa. Količnik smo spreminjali po stopnjah 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000. Pri tej raziskavi smo izbrali graf iz tiste podmnožice G^* grafov množice G, za katero velja:
1., da je $G^* \gg \rceil\ G^*$, in
2., da so grafi iz množice G^* v prvi fazi raziskave dali podobne rezultate.

Tretjo fazo raziskave smo posvetili izbiri najprimernejšega načina razporejanja procesov pri **dopustnem količniku prenosnega časa** v odvisnosti od dveh kriterijev:

- časa izvajanja HADFG
- učinkovitosti procesorskega polja

4.3.1. DOPUSTNI KOLIČNIK PRENOSNEGA ČASA

Pri dani konfiguraciji polja je čas izvajanja grafa HADFG poleg algoritmičnih lastnosti grafa HADFG tudi funkcija razmerja T_{izv}/T_p. Normirani čas izvajanja grafa HADFG je razmerje med tem časom pri izbranem razmerju T_{izv}/T_p in med minimalnim časom izvajanja.

Dopustni količnik prenosnega časa je tisto razmerje T_{izv}/T_p, pri katerem je normirani čas izvajanja grafa enak dvakratni vrednosti minimalnega časa izvajanja.

4.3.2. ČAS IZVAJANJA HADFG

Čas izvajanja grafa HADFG smo vzeli kot merilo za primerjavo med različnimi načini razporejanja. Ker je ta čas odvisen od algoritmičnih lastnosti grafa HADFG in od konfiguracije procesorskega polja, smo definirali vsoto $S_k\left(G_j\right)$, ki je vsota časov iz-

vajanj HADFG iz izbrane množice algoritmov pri določeni konfiguraciji procesorskega polja. Velja:

$$S_k\left(G_j\right)=\sum_i T_i$$

kjer je:

k ... element iz množice konfiguracij {2x2, 3x3, 4x4}
G_j ... element iz množice $\{G_s, G_n\}$
T_i ... čas izvajanja i -tega grafa HADFG iz množice G_j. V našem primeru je i element iz množice $\{1,2,...,20\}$.

Primerjava vsot $S_k\left(G_j\right)$ pri različnih načinih razporejanja nam daje oceno uspešnosti posameznih načinov.

4.3.3. UČINKOVITOST PROCESORSKEGA POLJA

Učinkovitost procesorskega polja E je razmerje med ceno izvajanja grafa HADFG na enem procesorju in ceno izvajanja na večprocesorskem sistemu. Idealna vrednost je 1, kar pomeni, da je paralelni sistem maksimalno izkoriščen.

$$E=\frac{T_s}{n\cdot T'}$$

kjer je:

n ... število procesorjev v sistemu
T' ... povprečen čas zasedenosti procesorjev
T_s ... čas sekvenčnega izvajanja grafa HADFG.

Čas sekvenčnega izvajanja grafa HADFG je tisti čas, ki je potreben, da se graf izvede na enem samem procesorju. Torej je:

$$T_s=\sum_i T_{pi},$$

kjer je T_{pi} čas izvajanja i-tega procesa grafa HADFG.

4.3.4. ZASEDENOST PROCESORSKEGA POLJA

Procesor je zaseden v dveh primerih:

- kadar izvaja proces, ki je določen z vozliščem grafa
- kadar pričakuje podatke ali rezultate (je blokiran)

Naj bo T_{izvj} čas izvajanja procesov na i-tem procesorju in T_{blokj} čas blokiranosti i-tega procesorja polja.

Čas zasedenosti procesorskega polja T_{zas} je podan z enačbo

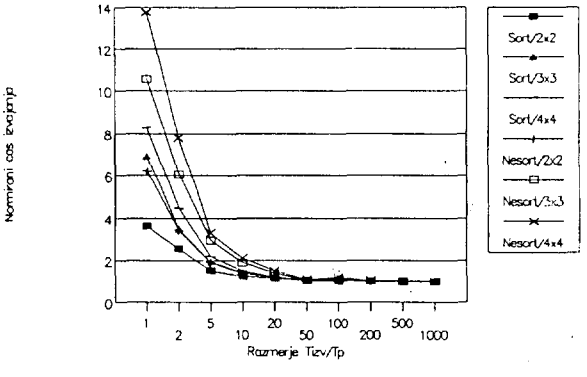
$$T_{zas}=\sum_i T_{izvi}+\sum_i T_{blokj}$$

Zasedenost procesorskega polja Z je razmerje med časom povprečne zasedenosti procesorjev v polju in časom izvajanja grafa HADFG na tem istem polju.

$$Z=\frac{T'_{zas}}{T_{graf}},$$

kjer je:

T_{graf} ... čas izvajanja grafa HADFG na procesorskem polju, in
T'_{zas} ... čas povprečne zasedenosti procesorjev v polju



Slika 3. Normirani čas izvajanja za različne K

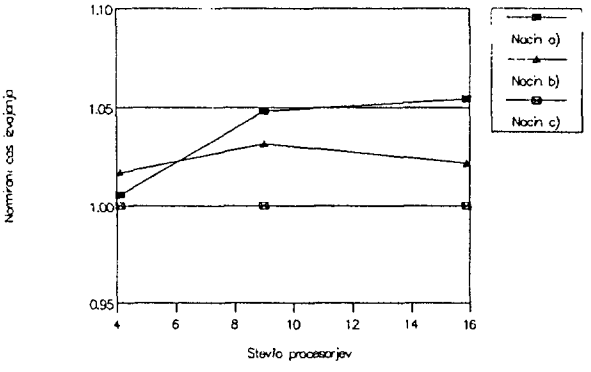
Povprečna zasedenost procesorjev v polju je:

$$T'_{zas}=\frac{T_{zas}}{n},$$

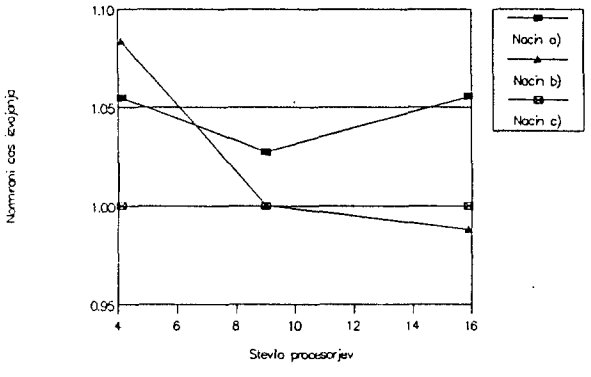
kjer je T_{zas} čas zasedenosti procesorskega polja in n število procesorjev.

5. REZULTATI SIMULACIJE

Simulacija je pokazala, da je najbolj primerna kvadratna oblika procesorskega polja. Takšna oblika omogoča v povprečju krajše razdalje med procesorji in manjšo obremenitev prenosnih poti. Zato smo se pri nadaljnih raziskavah omejili na kvadratna procesorska polja velikosti 2x2, 3x3 in 4x4.

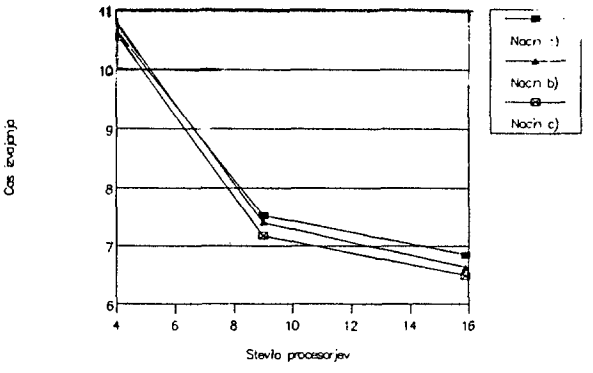


a) sortirani grafi

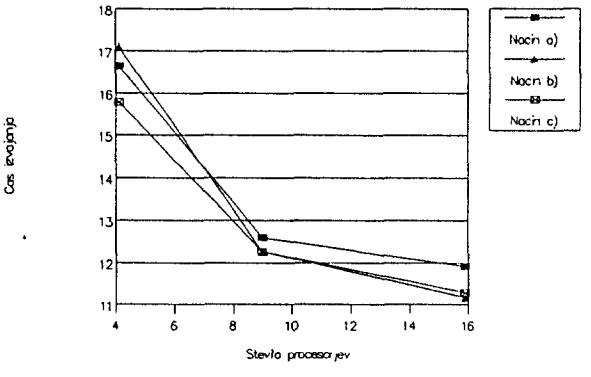


b) nesortirani grafi

Slika 4. Primerjava po normiranem času izvajanja



a) sortirani grafi

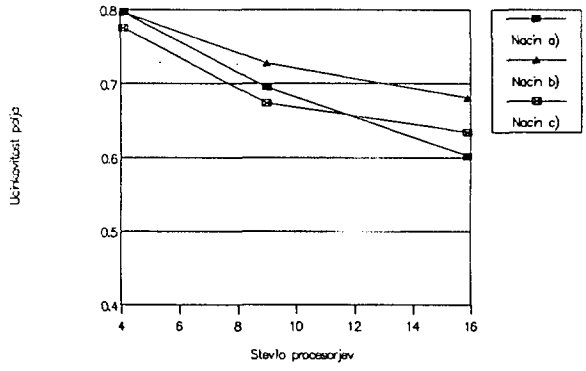


b) nesortirani grafi

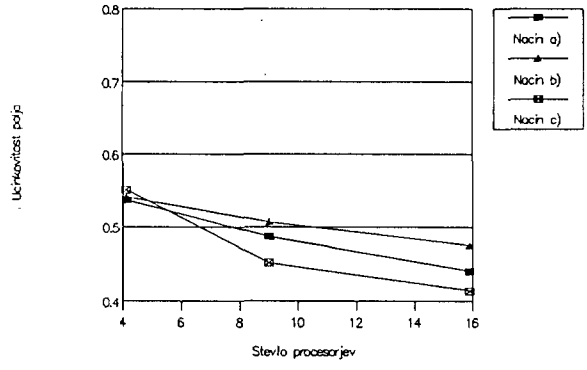
Slika 5. Primerjava po času izvajanja

Iz slike 3 je razvidno, da se dopustni količnik prenosnega časa spreminja z velikostjo procesorskega polja. Manjše procesorsko polje zahteva nižji, večje polje pa višji dopustni količnik časa prenosa. Dopustni količnik za polje 4x4 za sortiran graf je 6 in za nesortiran graf je 12. Pri nadaljnih simulacijah smo se omejili na količnik 10.

Slika 4 prikazuje normirano vsoto, ki je podana z razmerjem med $T_k\left(G_j\right)$ pri izbranem načinu razporejanja in $T_k\left(G_j\right)$ pri dvonivojskem načinu razporejanja, v odvisnosti od konfiguracije procesorskega polja. Dvonivojski način razporejanja daje v pogledu časa izvajanja najboljše rezultate za konfiguraciji 2x2 in 3x3. Majhno število procesorjev na prvem nivoju razporejanja preprečuje pri enonivojskem načinu širjenje procesov po celotnem polju, medtem ko lahko dvonivojski način razporejanja uporabi vse nezasedene procesorje. Zanimivo je, da na polju 2x2 kaže najslabše rezultate enonivojski način razporejanja z dvostranskim prenosom. Predvidevamo, da so takšni rezultati posledica dodatnega zmanjšanja moči procesiranja zaradi istočasnih dvostranskih prenosov. Dvonivojsko razporejanje pri konfiguraciji 4x4 in nesortiranih grafih daje presenetljivo slab rezultat, kar je posledica dvonivojskega razdeljevanja večjih poddreves.



a) sortirani grafi



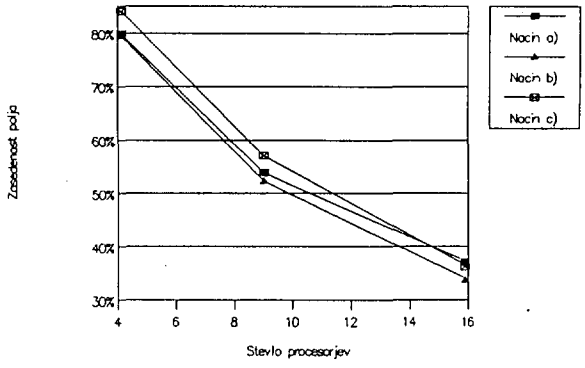
b) nesortirani grafi

Slika 6. Primerjava po učinkovitosti polja

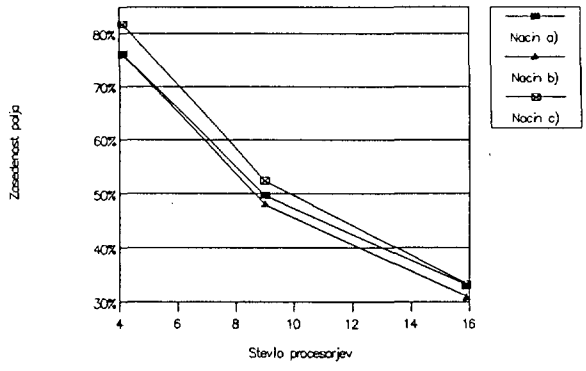
Sortirani grafi imajo paralelne veje razporejene tako, da se veje, ki zahtevajo daljše prenosne čase, izvajajo na bližjih procesorjih, kar občutno zmanjša prenosne čase. Primerjava med vsotami T_k za sortirane in nesortirane grafe na sliki 5 kaže, da se sortirani grafi izvajajo neprimerno hitreje od nesortiranih.

Učinkovitost polja kaže slika 6. Enonivojski način razporejanja s prenosom podatkov po dveh zankah je najbolj učinkovit, dvonivojski način pa najmanj. Izjemoma je pri dvonivojskem načinu razporejanja, nesortiranih grafih, in konfiguraciji 2x2 učinkovitost polja večja.

Zasedenost polja kaže slika 7. Največja zasedenost se pojavlja pri dvonivojskem in najmanjša pri enonivojskem načinu razporejanja s prenosom podatkov po dveh zankah hkrati.



a) sortirani grafi



b) nesortirani grafi

Slika 7. Primerjava po zasedenosti polja

6. ZAKLJUČEK

Članek predstavlja poskus verifikacije hierarhičnega večprocesorskega sistema, ki je opisan v delu /6/. Verifikacija se opira na simulacijo sistema. Izdelana je analiza delovanja transputerskega polja v odvisnosti od načina razporejanja, velikosti polja in količnika prenosnega časa.

Potrjeno je dejstvo, da imajo razdalje med procesorji pomembno vlogo pri prenašanju sporočil. Analiza

sistema je pokazala, da je mogoče doseči s krajšimi zankami in večjim številom možnih komunikacijskih poti boljše rezultate. Zato je kvadratna oblika procesorskega polja praviloma boljša od pravokotne oblike.

Količnik prenosnega časa naj bo za uspešno izvajanje čim večji, medtem ko je velikost dopustnega količnika ocenjena na 10. Ta količnik pogojuje stopnjo granulacije algoritma, ki se izvaja na transputerskem sistemu. Pri količniku, ki je večji od dopustnega, postane učinkovitost sistema bistveno večja.

Obravnavani sistem omogoča več načinov dinamičnega dodeljevanaja procesov. Raziskave učinkovitosti treh predlaganih načinov so dale naslednje rezultate:

- Enonivojski način razporejanja procesov s prenosom podatkov po eni zanki je najmanj učinkovit; je najpočasnejši in slabo izkorišča procesorsko polje.
- Enonivojski način razporejanja procesov s prenosom podatkov po dveh zankah hkrati ima največjo učinkovitost in najbolje izkorišča dano procesorsko polje, vendar ni tudi najhitrejši. Priporočljiv je v primerih, ko izvajamo na enem procesorskem polju več opravil hkrati in v polje vstopamo sočasno skozi več procesorjev. Izkoristek procesorskega časa je tu najboljši.
- Najhitrejše je izvajanje opravil pri dvonivojskem načinu razporejanja procesov s prenosom podatkov po dveh zankah. Cena, ki jo zahteva večja hitrost, se kaže v manjši učinkovitosti procesorskega polja.

7. LITERATURA

/1/ L.C.Waring, A general purpose communications shell for a network of transputers, North-Holland, Microprocessing and Microprogramming Vol.29 (1990), 107-119

/2/ A.Ciampolini, A.Corradi, L.Leonardi, Parallel object system support on transputer-based architectures, North-Holland, Microprocessing and Microprogramming Vol.27 (1989), 339-346

/3/ C.Jesshope, Parallel processing, the transputer and the future, Butterworth & Co. (Publishers) Ltd., Microprocessors and Microsystems, Vol.13 No.1, 1989, 33-37

/4/ INMOS Spectrum, "Product information, The Transputer Family", March 1988.

/5/ C.Jesshope, Transputers and switches as objects in OCCAM, North-Holland, Parallel Computing 8 (1988), 19-30

/6/ P.Kolbezen, P.Zaveršek, Hierarhični večprocesorski sistem, Informatica, A Journal of Computing and Informatics, Vol.15, Nr.1, Feb. 1991, 65-76

/7/ C.Barmon, M.N.Faruqui, G.P.Battacharjee, Dynamic load balancing algorithm in a distributed system, North-Holland, Microprocessing and Microprogramming, Vol.29 (1990), 273-285

/8/ U.De Carlini, U.Villano, The routing problem in transputer-based parallel systems, Butterworth-Heinemann Ltd., Microprocessors and Microsystems, Vol.15 No.1, 1991, 21-33