

ZORAN JOVANOVIĆ

**INSTRUKCIJSKI NIVO
PARALELIZMA**

BEOGRAD 2006

CIP Katalogizacija u publikacija
Narodna biblioteka Srbija, Beograd

004.41.032.24

Jovanović, Zoran
Instrukcijski nivo paralelizma / Zoran Jovanović
Beograd : ATC Avangarda,
2006 (Beograd : ATC Avangarda).
142 str. : grafički prikazi : 30

Tiraž 500. – Bibliografija: str. 139-142.

ISBN 86-85117-12-7

a) Softversko inženjerstvo – Paralelizam
COBISS.SR-ID 136865292

Recenzent: profesor dr Dragan Milićev

**SUPRUZI IDI
SINOVIMA BOJANU I VOJINU
RODITELJIMA DARINKI I MITI
NEKADAŠNJOJ LABORATORIJI
RT270 INSTITUTA VINČA**

SADRŽAJ

Predgovor

0. Uvod o paralelizmu	1
1. Uvod u instrukcijski nivo paralelizma	3
1.1. Paralelizam na instrukcijskom nivou i mrežno planiranje	3
1.1.1. Mrežno planiranje i paralelizacija kôda na nivou instrukcija.	3
1.2. Grafovi zavisnosti po podacima i tipovi zavisnosti unutar bazičnog bloka	5
1.2.1. Prave zavisnosti po podacima	5
1.2.2. Antizavisnosti po podacima	8
1.2.3. Izlazne zavisnosti po podacima	10
1.2.4. Grafovi zavisnosti po podacima u optimizaciji kôda	10
1.3. Lokalna optimizacija kôda za idealnu i realnu mašinu	11
1.3.1. Idealna mašina	11
1.3.2. Najranije izvršavanje	11
1.3.3. Ostali optimalni rasporedi na idealnoj mašini	12
1.3.4. Eksponencijalnost algoritma za optimalan kôd za realne mašine	12
1.4. Heuristički algoritmi	15
1.5. Limiti u paralelizmu kada se optimizacija sprovodi samo na nivou bazičnih blokova.	18
2. Paralelizam u kôdu bez programskih petlji	20
2.1. Prevazilaženje limita u paralelizmu posmatranjem globalnog programa	20
2.2. Osnovne selidbe operacija između bazičnih blokova	21
2.2.1. Selidbe preko spoja	22
2.2.2. Selidbe preko grananja	23
2.2.3. Selidbe ciklusa operacije	25
2.3. Kontrolne zavisnosti i programski grafovi zavisnosti	25

3. Strategije izbora selidbi u cilju povećanja paralelizma	28
3.1. Trace scheduling	29
3.1.1. Izbor najverovatnijeg neraspoređenog traga	30
3.1.2. List scheduling	31
3.1.3. Obnavljanje granica i spojeva na optimizovanom tragu	32
3.1.4. Selidba Skokova	37
3.1.5. Generalnija pravila selidbi	38
3.1.6. Pomoćni blokovi – Pojavljivanje i eliminacija blokova	38
3.1.7. Eksplozija kôda	38
3.2. Optimizacija kôda petlji primenom	
Trace scheduling-a – razmotavanje petlji	39
3.2.1. Broj iteracija petlje poznat u vreme prevođenja	40
3.2.2. Broj iteracija petlje nije poznat u vreme prevođenja	41
3.2.3. Operacije invarijantne u odnosu na petlju	41
4. Spekulativno izvršavanje	42
4.1. Eliminacija kontrolnih zavisnosti	43
4.1.1. Eliminacija kontrolnih zavisnosti prilikom prevođenja	44
4.1.2. Eliminacija kontrolnih zavisnosti u vreme izvođenja	54
4.2. Tronivoski prediktor grananja	57
5. Protočnost, resursi mašine i paralelizam	61
6. Zavisnosti po podacima u programskim petljama	63
6.1. Granice petlje	63
6.2. Generalizacija problema zavisnosti kod petlji i sistemi	
Diofantskih jednačina	65
6.2.1. GCD test	66
6.2.2. Banerjee test	68
6.2.3. Totalni test	69
6.2.4. Redukovani test	69
6.3. Opšti slučaj višedimenzionog niza u dve ugnježdene petlje	70

7. Optimizacija petlji - softverska protočnost	72
7.1. Fiksne iteracione distance	72
7.1.2. Obeležavanje grana grafa zavisnosti po podacima programske petlje za fiksne iteracione distance	73
7.1.3. Ugnježdene i nadgnježdene petlje	75
7.2. Grafovi zavisnosti po podacima za programske petlje	75
7.2.1. DoAll i DoAcross petlje	77
7.2.2. Grafovi razmotane petlje za DoAll i DoAcross	79
7.3. Softverska protočnost preko smicanja susednih iteracija	82
7.4. Opšti model ulaska u novu iteraciju kod softverske protočnosti	90
7.5. Generalizacija softverske protočnosti u grafu beskonačno razmotane petlje	93
7.5.1. Ograničenja nove iteracije kod softverske protočnosti	94
7.5.2. Uslovi da nova iteracija bude semantički ispravna.	96
7.6. Ograničenja u brzini izvršavanja za DoAcross petlje	100
7.7. Optimizacija petlje kod softverske protočnosti	103
7.7.1. Komponente jake povezanosti u grafu petlje	105
7.7.2. Promene iteracionih distanci za inter SCC grane grafa	106
7.7.3. Paralelizam u acikličkom grafu jedne iteracije kada je primenjena protočnost komponenti jake povezanosti	107
7.7.4. Ukupan postupak optimizacije kôda primenom protočnosti komponenti jake povezanosti	120
8. Procesori koji koriste paralelizam na instrukcijskom nivou	122
8.1. Rani RISC procesori	122
8.2. Dataflow procesori	125
8.3. VLIW procesori (EPIC – Explicit Parallel Instruction Computers)	125
8.4. Superskalarni procesori	126
8.4.1. Uslovni skokovi i popunjavanje instrukcijskog prozora	128
8.4.2. Određivanje zavisnosti i preimenovanja	129
8.4.3. Stanja instrukcija	130
8.4.4. Oporavak od izuzetaka i greške u predikciji grananja	132
8.5. Odnos optimizacije u prevođenju i run time paralelizacije	

superskalarnih procesora	137
8.6. Današnji procesori	137
8.7. Zaključak	138
9. Literatura	139

PREDGOVOR

Monografija je rezultat rada u oblasti paralelizma započetog u izuzetnom razvojnom timu Laboratorije za računarsku tehniku RT270, Instituta Vinča. Bilo je to zlatno doba instituta u Srbiji, tokom koga su se naši razvojni timovi u praktičnim i teorijskim znanjima u računarstvu takmičili sa vodećim svetskim silama. Danas je preko 60% tog razvojnog tima iz RT270, koji je vodio dr Miodrag Čarapić, u Kanadi, SAD, Australiji i mnogim drugim zemljama sveta. Uspeh u svetu (više od 7 profesora univerziteta u SAD i Kanadi i vodeće pozicije u razvojnim laboratorijama vodećih svetskih kompanija) je svedok kvaliteta tog ekstremno sposobnog tima. Ne očekujem da će se lako ponovo okupiti takav skup talenata na jednom mestu u institutima u Srbiji, mada mi je želja da se to dogodi. Na žalost, tim RT270 osuo se krajem osamdesetih i početkom devedesetih godina prošlog veka.

Nakon dve godine od zapošljavanja u Vinči, dobio sam zadatak da vodim razvoj array procesora, da bi se ubrzo formirao tim koji su činili još Jovan Ognjanović, Darko Mušicki, Zorica Mihajlović i Radovan Milunov. Simulirani procesor sa tada neverovatnih 30 MFLOPSa bio je završen (i samim tim projektovan procesor) pred Novu 1982. godinu. Taj simulator razvijen je u programskom jeziku PASCAL i zajedno sa paralelnim assemblerom, koji je razvio Darko Mušicki, pomogao je da ceo tim pronikne u tajne paralelizma. Kompletan softver za sprezanje sa multiprocesorskim računarom domaćinom uradio je Jovan Ognjanović. Ključnu ulogu u programiranju imao je Radovan Milunov, ali je bilo angažovano i nekoliko najtalentovanijih studenata zadnje godine studija matematike. Najveći doprinos od njih su dali Zsolt Zolnay koji radi fundamentalna istraživanja u oblasti magnetne rezonance na Mayo klinici i Zoran Obradović, profesor univerziteta u SAD, sa kojim i danas saradujem. Na kraju, 1985. godine razvijena su tri prototipa procesora nazvanog FRT300, korišćenjem specijalnih prototipskih wire-wrap ploča za tadašnje najbrže tehnologije logičkih kola. U to doba nismo još znali da će se procesori kao FRT300 zvati danas VLIW ili EPIC procesorima. Njihovom današnjem uspehu doprinele su nove tehnologije programskih prevodilaca kojima je rešena automatska paralelizacija kôda.

Apsurd je da procesor FRT300 nije mogao odmah da se proizvede u tehnologiji višeslojne štampe, jer Jugoslavija još nije imala takvu tehnologiju. Na ETANu na Bledu demonstriran je kompletan razvoj i rad procesora FRT300, od hardvera, preko svih komponenti razvojnog, sistemskog i dijagnostičkog softvera, do algoritama za obradu signala i dvodimenzionu Furijevu transformaciju koji se inicirao sa domaćina zasnovanog na tada ultra modernom procesoru 80286. Procesor FRT300 je u izvršavanju demonstriranih algoritama bio za preko 100 puta brži od VAX11/785 procesora, tada najbržeg mini računara u zemlji.

Postepeno gašenje najvažnijih projekata u Laboratoriji RT270, otvaranje nastavničkog mesta na Elektrotehničkom fakultetu u Beogradu, osipanje tima iz Laboratorije i trend da optimizacija koda u programskim prevodiocima paralelnih mašina postaje centralni problem, dovodi do druge orijentacije u istraživanjima i razvojima tokom devedesetih. Nastavili smo da se na Elektrotehničkom fakultetu u Beogradu bavimo paralelizmom, a nekoliko generacija studenata imalo je priliku da korišćenjem paralelnog assemblera

programira simulirani procesor FRT300. U prvom periodu ključni saradnik na tim istraživanjima bio je Dejan Petković, koji je tadašnjim generacijama studenata ostao u uspomeni kao izuzetan predavač. Meni su ostale najlepše uspomene na zajedničke analize različitih tehnika optimizacije koda, pre svega kada smo zajednički savladavali tajne spekulativnog izvršavanja i u svom razumevanju bili veoma blizu vodećim istraživačima u svetu. Dejan i ja smo vodili oko 10 diplomskih radova u kojima su detaljno analizirani razni aspekti instrukcijskog nivoa paralelizma. Dejan je pokrenuo realizaciju paralelnog programskog prevodioca i na prvoj verziji završio svoju magistarsku tezu. To je značilo pripremu za zaokruživanje eksperimentalnog okruženja na kome bi se u paralelizaciji kalile nove generacije pametnih studenata ETFa u Beogradu. Tada je došao rat i mobilizacije, a mi smo i dalje bili u svom svetu paralelizma. Na žalost svih nas koji smo sa njim sarađivali, Dejan je rešio da ode u svet, a sa njim i deo poleta ekipe.

Urađeno je nekoliko magistarskih radova iz instrukcijskog nivoa paralelizma, a posebno bih istakao radove Aleksandra Lazarevića i Nenada Teofilovića koji su bili posvećeni paralelizmu programskih petlji. Sudbina se poigrala tako da je kasnije Aleksandru Lazareviću mentor za doktorat bio profesor Zoran Obradović, jedan od studenata matematike angažovanih mnogo godina ranije na razvoju procesora FRT300.

Ponovni zamah u radu na instrukcijskom nivou paralelizma nastao je sa uključivanjem Dragana Milićeva u istraživanja. Radili smo u oblasti paralelizma petlji sa uslovnim grananjima, jednoj od oblasti instrukcijskog nivoa paralelizma u kojoj su teorijska znanja najmanja. Ova monografija ne obuhvata navedeni problem, jer se njemu može posvetiti jedna kompletna monografija, a najvažnije ideje u oblasti modeliranja te vrste petlji su potekle od Dragana Milićeva. Model razvijen tokom tih istraživanja verovatno je i danas najkompletniji model za petlje sa uslovnim grananjima na svetu.

Monografija pokušava da premosti jaz koji postoji u razumevanju paralelizma između onih koji su posvećeni hardveru i softveru u oblasti. Jaz je premošćen pomoću jedinstvenog modela, zasnovanog na matematičkom aparatu teorije grafova. Tek kada su elementi paralelizma analizirani i objašnjeni do granica limita modela, prešlo se na implementaciju paralelizma. Tako su objašnjene implementacije u vreme prevođenja i u vreme izvršavanja. Nadam se da će knjiga pomoći čitaocima da ovladaju tajnama današnjih modernih procesora i optimizacionih delova modernih programskih prevodilaca.

Posebno bih hteo da istaknem saradnju sa studentima poslediplomskih studija i pete godine Elektrotehničkih fakulteta koji su dobijali ranije verzije ovog rukopisa i koristili ga za spremanje ispita iz paralelnih računarskih sistema. Davali su mi signale na kojim mestima treba dopuniti i popraviti tekst, postavljajući pitanja u delovima u kojima im nije bio potpuno jasan tekst. Davali su sugestije kako popraviti razumljivost rukopisa. Zahvalan sam im na toj pomoći. Tako je nekoliko generacija pomoglo da naredne generacije imaju bolje tekstove na raspolaganju. Do sada je ova materija korišćena u naprednim kursevima na Elektrotehničkim fakultetima u Beogradu i Banja Luci, a posle pojave knjige nadam se da će se koristiti i na drugim mestima.

Autor