

Keywords: parallel algorithms, measuring of parallelism, task graph, simultaneous degree, coupled degree

Borut Jereb  
Ljubo Pipan\*  
Institut J. Stefan  
Fakulteta za elektrotehniko in  
računalništvo\*, Ljubljana

Paralelne algoritme lahko predstavimo z grafom opravil. Iz grafa opravil je možno enostavno razbrati stopnjo sočasnosti in stopnjo povezanosti algoritma, ki ga graf predstavlja. Ti stopnji predstavljata pomembni značilnosti paralelnih algoritmov.

V tem članku je podan predlog za izračun mere sočasnosti in mere povezanosti. Opisane so tudi nekatere značilnosti obeh mer.

Izračun mer je neodvisen od arhitekture, na katerem bo tekel algoritem; sami meri pa lahko predstavljata velikost vložka, ki je potreben za sprogramiranje algoritma.

#### MEASURING THE PARALLELISM

Parallel algorithms could be represented by the task graph, which is at the same time a simple representation of the simultaneousness degree and coupled degree of an algorithm, represented by the graph. The above mentioned degrees are most important features of the parallel algorithm.

This study suggests a method for calculating the simultaneousness degree and the coupled degree. Included are some of their main characteristics.

The calculation of the degrees does not depend on the architecture within which the algorithm will operate; the degrees themselves though represent the size of the insert required for the algorithm to be programmed.

#### UVOD

Ko začnejo človeka zanimati paralelne arhitekture, paralelno procesiranje, ... in sploh vse, kar naj bi bilo paralelnost, vzame v roko literaturo in prične s študiranjem le te. Literatura v obliki knjig in člankov je ogromno, saj je to že vrsto let popularna tema, ki je še vedno do neke mere ovita v plašč misterioznosti in zato tem bolj buri duhove. Pri prebiranju literature si ustvarjamo svojo predstavo o stvareh, ki so predmet avtorjevih razmišljanj. Eni takšno, drugi drugačno - pač odvisno od osebe.

No vrnimo se nazaj k paralelnosti. V literaturi, ki obravnava paralelizem je ničkolikokrat omenjena SOČASNOST in medsebojna odvisnost ali POVEZANOST v zvezi z opravili, procesi, instrukcijami. Ne mislim se spuščati v definicijo(e) paralelnosti, saj jih je veliko, če ne celo preveč; vsakemu pa je intuitivno jasno kaj to je. Podobno je s sočasnostjo in povezanostjo. In

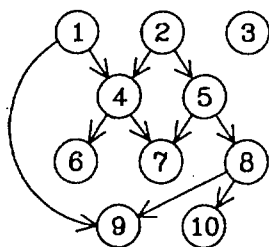
kaj je z mero za sočasnost in mero za povezanost? V tem članku bom skušal podati ti meri. Naj se imenujeta stopnja sočasnosti in stopnja povezanosti.

Kakor je ob praznih kozarcih brezpredmetno govoriti o kvaliteti vina, ki naj bi bilo v kozarcih, tako je brezpredmetno govoriti o bolj ali manj paralelnih algoritmih. V tehniki je potrebno pojme oceniti. Ocena pojma je dobljena po točno določenem algoritmu in je v veliki večini primerov predstavljena s števili iz intervala, ki jih ta ocena lahko zavzame.

Postavlja se vprašanje, kako oceniti paralelizem, ki smo ga izbrali oziroma določili v nekem algoritmu. Sem prepričan, da mera za sočasnost in mera za povezanost programskih modulov, dajeta dobro osnovo za oceno paralelizma v poljubnem algoritmu. Pri tem sta ti meri neodvisni od arhitekture na kateri se naj bi izvajal algoritem in s tem daje splošno primerljive rezultate.

## PREDSTAVITEV PARALELNEGA ALGORITMA

Najprej bo opisan način predstavitve paralelnega algoritma (način določevanja paralelnih modulov in določevanje njihove medsebojne - večkrat podatkovne odvisnosti - naj nas ne zanima). Paralelni algoritem naj bo predstavljen z grafom. Takšna predstavitev je v literaturi splošno uporabljana.<sup>1,2,3,4,6</sup>



Slika 1 Predstavitev paralelnega algoritma z grafom opravil.

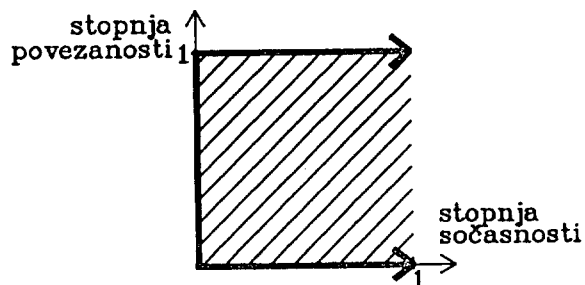
Ker je paralelni algoritem predstavljen z grafom, si oglejmo lastnosti grafa tako, da jih bomo primerjali z lastnostmi paralelnih algoritmov. Vozlišče grafa predstavlja programski modul. Vsi programski moduli, pri katerih je mišljeno sočasno izvajanje, imajo isti nivo. Nivo je utež vozlišča. Pri tem velja, da ležijo na nivoju ena vsi tisti programski moduli, ki se izvajajo sočasno in so neodvisni med seboj in neodvisni od ostalih programskih modulov. Sočasno izvajajoči programski moduli, ki so med seboj neodvisni in odvisni od programskih modulov na nivoju ena, ležijo na nivoju dva. Nivo tri predstavlja množico, med seboj neodvisnih programskih modulov, ki so odvisni od programskih modulov na obeh nižjih nivojih in se izvajajo sočasno; itd. Odvisnost med programskimi moduli je v grafu predstavljena z usmerjenimi povezavami med vozlišči grafa. Povezava poteka vedno v smeri od vozlišča, ki leži na nižjem nivoju, k vozlišču na višjem nivoju. Pri tem lahko ostanejo nekatera vozlišča v grafu nepovezana. Graf je acikličen, saj lahko ciklično izvajanje nekega programskega modula predstavimo s podvajanjem tega programskega modula. Definirajmo še višino grafa, ki naj bo enaka največjemu nivoju, ki ga ima katerokoli od vozlišč grafa.

Rekurzivno izvajanje programskih modulov rešimo s podvajanjem vozlišča, ki predstavlja rekurzivni programski modul.

Za primer predstavitve nekega paralelnega algoritma z grafom, si oglejmo Sliko 1, ki predstavlja algoritem, razdeljen na deset modulov. Pri tem je možno sočasno izvajanje modulov, predstavljenih z vozlišči 1, 2 in 3, ki ležijo na nivoju ena. Nivo ena vsebuje tudi nepovezan programski modul, oziroma izolirano točko. Na nivoju dva sta dva medsebojno neodvisna sočasno tekoča programska modula, ki sta odvisna od modulov prejšnjega nivoja. Ker obstajata še dva nivoja, je višina grafa enaka štiri.

## DEFINICIJI MER

Najprej bom opisal vrednosti, ki jih ti dve meri lahko zavzameta. Obe naj bosta iz množice racionalnih števil na intervalu med nič in ena. In kako dosežemo skrajni točki pri obeh merah? Mera za sočasnost naj bo enaka nič, ko je algoritem popolnoma sekvenčen (zaporeden). Vrednost ena pa naj zavzame takrat, ko je možno razbiti algoritem na neskončno modulov, ki se bodo izvajali v istem časovnem intervalu, oziroma trenutku. Mera za povezanost naj bo enaka nič, ko so vsi programski moduli med seboj neodvisni in enaka ena, ko je vsak programski modul odvisen od vseh predhodno izvedenih programskih modulov, pri popolnoma sekvenčnem programu.



Slika 2 Področje v katerem je definirana dvojica (stopnja sočasnosti, stopnja povezanosti).

Če so lastnosti paralelnih algoritmov predstavljene z dvema številoma, lahko dvojica (stopnja sočasnosti, stopnja povezanosti) zavzame, glede na vse dosedaj povedano, vse vrednosti, ki ga določa šrafirano področje in poudarjena črta na sliki 2.

Definirajmo:

1. Stopnja sočasnosti:

$$CD = \frac{n - TH}{n}$$

pri čemer je  $n > 1$  in predstavlja število vozlišč grafa.  $TH$  predstavlja višino drevesa.

2. Stopnja povezanosti

$$SD = \frac{EN}{\frac{n(n-1)}{2}}$$

pri čemer je  $n > 1$  in predstavlja število vozlišč grafa.  $EN$  predstavlja število povezav v grafu.

Iz teh dveh definicij sledi nekaj trditev:

1. Maksimalno stopnjo sočasnosti dobimo v limiti, ko smo problem razdelili na neskončno programskih modulov, ki se izvajajo sočasno. Limita stopnje sočasnosti je takrat enaka 1.
2. Minimalno stopnjo sočasnosti dobimo pri popolnoma sekvenčno organiziranem algoritmu in je enaka 0.
3. Pri  $SD = 0$  je možno vse module paralelnega algoritma izvajati obenem. Velja tudi obratno.

4. Pri  $CD = 0$  je  $\frac{2}{n} \leq SD(n) \leq 1$ .  
 5. Pri dani stopnji sočasnosti je spodnja meja za mero povezanosti enaka  $\frac{T_H - 1}{(n-1)n}$ .

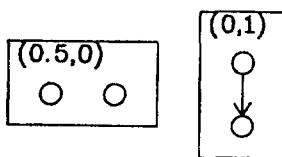
Z definicijo neke nove mere na osnovi pravkar definiranih stopenj bi izgubili na enostavnosti in jasnosti predstavitve s tema dvema, za paralelizem pomembnima lastnostima paralelnih algoritmov.

### LASTNOSTI DVOJICE ( $CD$ , $SD$ )

Vrednosti obeh mer dajeta slutiti obliko grafa, ki predstavlja algoritem. Potrebno je tudi povedati, da smo pri tem izpustili dodatno utežitev vozlišč, ki bi predstavljala čas izvajanja. Čas bi lahko bil absoluten, izražen v časovnih enotah ali relativen glede na čas izvajanja nekega opravila v sistemu. Takšna merjenja so v literaturi obdelana.<sup>2</sup> So zahtevnejša, ter nemalokrat zavisijo od podatkov in ostalih težko predvidljivih situacij, ki nastanejo med izvajanjem programa. Vrednosti mer za algoritem, ki ga predstavlja eno samo vozlišče ni definirano.

Pred točnim definiranjem obeh mer smo omejili področje, ki ga ti dve meri lahko zavzameta. Toda nekatere kombinacije  $CD$  in  $SD$  so nesmiselne. Na osnovi trditev in primerov si oglejmo vrednosti, ki jih lahko zavzame dvojica ( $CD$ ,  $SD$ ). Pri tem bomo pregledali vse možne vrednosti za  $CD$  in  $SD$ , pri razbitju algoritma na  $n$  modulov. Točke, ki bodo predstavljale mejne vrednosti, naj bodo povezane z daljicami.

Ker vrednosti mer za algoritem, ki ga predstavlja eno samo vozlišče, ni definirano, si najprej oglejmo primer, pri katerem smo algoritem razbili na dva modula. Slika 3 predstavlja vse možne načine izvajanja algoritma z dvema moduloma.

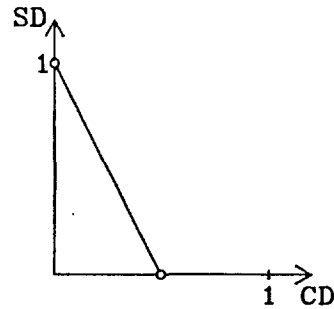


Slika 3 Vsi možni načini izvajanja paralelnega algoritma z dvema moduloma.

Vidimo, da sta možna le dva načina: popolnoma paralelna in popolnoma sekvenčen način izvajanja. Slika 4 predstavlja vrednosti, ki jih dvojica ( $CD$ ,  $SD$ ) lahko zavzame v  $CD - SD$  diagramu. Če mejni točki povežemo, dobimo med njima daljico.

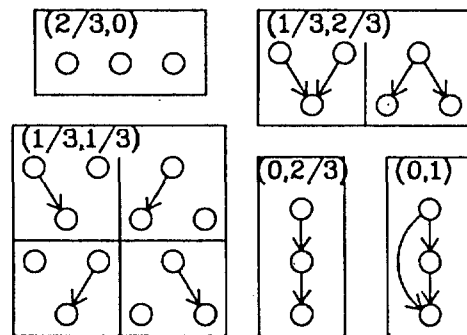
Iz primera je razvidno, da je pri popolnoma paralelnem načinu izvajanja algoritma mera za sočasnost enaka 0.5 in mera za povezanost enaka 0. Morda je na prvi pogled presenetljivo dejstvo, da ima 'popolnoma' paralelna algoritma stopnja sočasnosti enako le 0.5. Toda, če smo algoritem razbili na dva modula, je to čisto nekaj drugega kot, če smo ta isti algoritem uspeli razbiti na npr. 20000 med sabo popolnoma neod-

visnih modulov. Zgoraj definirana stopnja sočasnosti upošteva to razliko.



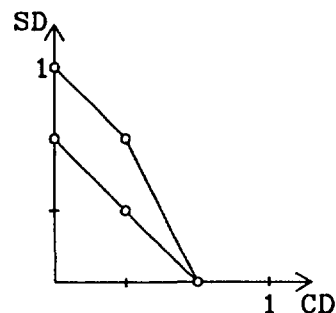
Slika 4 Daljica, ki povezuje obe možni vrednosti ( $CD$ ,  $SD$ ) paralelnega algoritma z dvema moduloma.

Kakšne so vse možnosti izvajanja in katere vrednosti lahko zavzame dvojica ( $CD$ ,  $SD$ ) pri razbitju algoritma v tri module, podajata sliki 5 in 6.



Slika 5 Vsi možni načini izvajanja paralelnega algoritma s tremi moduli

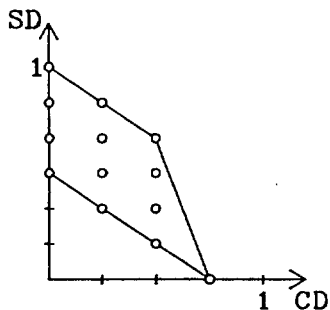
Če povežemo med seboj mejne točke, kot je to prikazano na sliki 6, dobimo nek lik.



Slika 6 Lik, ki nastane s povezovanjem vrednosti ( $CD$ ,  $SD$ ) paralelnega algoritma s tremi moduli.

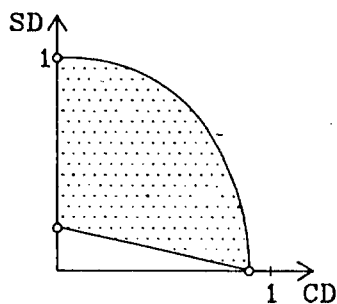
Slika 7 prikazuje možne vrednosti dvojic ( $CD$ ,  $SD$ ) v  $CD - SD$  diagramu pri razbitju algoritma na štiri module. Lik, ki ga dobimo s povezavo vseh mejnih

točk, spominja na lik, ki smo ga dobili v prejšnjem primeru.



Slika 7 Lik, ki nastane s povezovanjem vrednosti (CD,SD) paralelnega algoritma s štirimi moduli.

Slika 8 prikazuje možne vrednosti dvojic (CD,SD) pri razbitju algoritma na večje število modulov (deset). Vse možne vrednosti na sliki niso narisane, oblika lika pa približno podaja območje, v katerem se te vrednosti lahko nahajajo. Tudi z rastočim številom programskih modulov oblika lika ostaja podobna prejšnjima dvema.



Slika 8 Približen lik, ki nastane s povezovanjem vrednosti (CD,SD) paralelnega algoritma z desetimi moduli.

Na osnovi primerov smo se prepričali, da so mnoge kombinacije vrednosti stopnje sočasnosti in stopnje povezanosti nesmiselne.

Pravtako je iz primerov razvidno, da je oblika lika, ki ga dobimo s povezovanjem mejnih točk, podobna od primera do primera. Še posebej je razlika, ki nastopi pri različnem številu programskih modulov zelo majhna, ko sta ti števili veliki. Pri tem je treba izvzeti enostavni primer z dvema programskima moduloma, pri katerem se lik poenostavi v daljico.

Značilnost vseh likov je to, da imajo na CD osi (takrat je  $SD = 0$ ) le eno točko. Ta točka z večjim številom programskih modulov potuje od  $CD = \frac{1}{2}$  do  $CD = 1$ ; vendar slednje točke nikoli ne doseže.

Od tu gre meja po daljici, ki vsebuje spodnje mejne točke povezanosti, do točke na SD osi. Z večanjem števila programskih modulov, se najnižja točka na SD osi od vrednosti  $SD = 1$  približuje vrednosti  $SD = 0$ , ki jo pa nikoli ne doseže.

Vsi liki vsebujejo točko  $SD = 1$  pri  $CD = 0$ .

Za vse like je tudi značilno, da so zgornje mejne točke povezanosti nad premico  $SD = 1 - CD$ ; in to tem bolj, čim večje je število programskih modulov. Pri vseh  $CD > 0$  mejne točke nikoli ne dosežejo vrednosti  $SD = 1$ .

## SKLEP

Zgoraj opisani meri bosta morda v pomoč pri medsebojni komunikaciji, pri izražanju in pri razumevanju drugega človeka, kadar bo govor o sočasnosti in povezanosti poljubnega paralelnega algoritma. Ker opisujeta le graf opravil, ki smo ga določili na osnovi razbitja nekega algoritma, sta neodvisni od arhitekture ter časov izvajanja posameznih opravil - programskih modulov.

Algoritem za izračun obeh mer na osnovi danega grafa je zelo preprost, sam graf pa ima lastnosti, ki praktično ne predstavljajo omejitve pri opisovanju paralelnega algoritma.

## LITERATURA

1. K. Hwang and F. Briggs  
Computer Architecture and Parallel Processing  
McGraw Hill, New York, 1984
2. L. Kleinrock  
Distributed Systems  
IEEE Computer 18, 11 (Nov 85), 90-103
3. B. Kruatrachue and T. Lewis  
Grain size determination for parallel processing  
IEEE Software, jan 88, 23-32
4. J.L. Peterson and A. Silberschatz  
Operating system concepts  
Addison-Wesley pub. comp., 1984
5. D. Bajc in T. Pisanski  
Najnujnejše o grafih  
Presek, letnik 12, 84-85
6. S. Jajodia, J. Liu and P. A. Ng  
A scheme of parallel processing for MIMD systems  
IEEE Software engineering, jul 83, 436-445