

## 5. POSTUPAK OBLIKOVANJA PARALELNIH ALGORITAMA

- kvalitativno, u 6. kvantitativno

### 5.1. RACUNALNI I PROGRAMSKI MODEL

#### 5.1.1. RACUNALNI MODEL

- MMD računalo raspodijeljene mem.

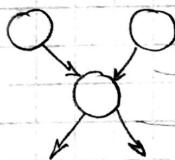
- brzina komunikacije: stalna (ne ovisi koja 2 proc. komuniciraju)

- lokalni pristup brži od udaljenog

- veće više PRAM-a

#### 5.1.2. PROGRAMSKI MODEL

- zadaci i komunikacijski kanali



- shelia: - po završetku posla, mora se poslati poruka

- prije početka posla, mora se primiti poruka (ACK)

- paralelni prog. se sastoji od više zad. koji se mogu izvoditi istovremeno

- ZADATAK je program koji izvodi lokalne operacije, stavlja i primavlja podatke, i sharauje novih podataka (sustav ne mora biti statican)

→ primanje je blokirajuće tako da uvisi potrebni učinkovi dodatni sincronizacijski mehanizmi

- svaki zad. slijedi proc., svih ne jedan proc....

- zadaci su proizvoljno pripunjeni procesorima i to ne utječe na rezultat

### 5.2. PRIMJERI PARALELNIH PROGRAMA

#### 5.3. FAZE OBLIKOVANJA PARALELNIH ALGORITAMA

① PODJELA = raspodjeljivanje problema (na zadatke)

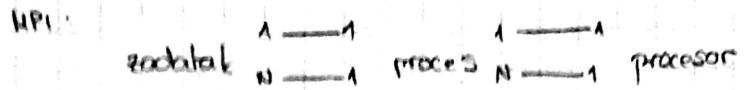
② KOMUNIKACIJA = određivanje minimalne potrebe komunikacije

③ AGLOMERACIJA = grupiranje zadataka ili komunikacijskih kanala u veće

④ PRIDRUŽUVANJE = dodjeljuvanje zadataka procesorima → NP težak problem

## 5.4. PODJELA

- cilj: postići situaciju da svaki procesor rabi podatke



- idealno: podjela koja ne zahtijeva dupliciranje podataka ili računanje

### a) PODJELA PODATAKA

- podaci se dijeli u manje cjeline koje obrađuju različiti zadataci

### b) PODJELA IZRAČUNAVANJA

- dijeli se izračunavanje, a ostalo o tome i podaci

## POŽELJNE SVOJSTVA

- broj zadataka barem za red veličine veći od broja procesora
- bez unesištačavanja podataka ili računanja
- zadataci predstavljaju veličine
- broj zadataka ovisi o veličini problema (upr. matrica  $n \times n \rightarrow n$  zadatake, a ne 100)
- \* razmotriti i sljedue algoritme

## 5.5. KOMUNIKACIJE

- cilj: raspodjeliti i zwaničiti vlastne kolice komunikacije

### PODJELA KOMUNIKACIJE

- lokalna / globalna

    - susjedna svima vest (svi uem, ja svemu)

- strukturirana / nestrukturirana

    - pravila struk

    (stabla, lanac, ...)

- statička / dinamička

    - zod. i kon.konali      mijenjaju se i konstaju  
    - ne mijenjaju      novi kon.konali

- sintrova / asintrova  $\Rightarrow$  plan

    - 2 zod. zwaju  
    do te komunicirati  
    komunikacija

    - ne zna se da li će biti  
    komunikacije  $\Rightarrow$  neplasti

## 5.5.1. LOKALNA KOMUNIKACIJA

\* korabijena metoda , G-S metoda - ne treba znati , ali treba biti sujestan da postoji

## 5.5.2. GLOBALNA KOMUNIKACIJA

pr. osmicanost na tlu - treba izračunati sve točke iznad \*

- konstanten komunikacijskih obrazaca

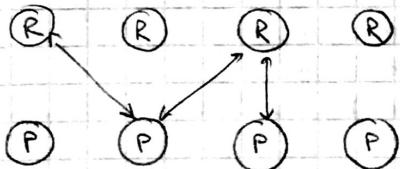
pr. reduciranje

## 5.5.3. ASINKRONA KOMUNIKACIJA

- primjer : obrada velike količine podataka

① raspodjela podataka po zadatacima uz obradu lokalnih i vanjskih zadataka

② posebni zadataci za računanje i zadataci zaduzeni za ažuriranje podataka



\* svatko se slobodno , 'ko god

\* MPI : probe f-je

## 5.5.4. KOMUNIKACIJA - POZELJNA SUOJSTVA

- podjednaka kol. komunikacije po zadatacima

- mala količina lokalne komunikacije

- istodobna komunikacija

- što manja oviznost zadataka

## 5.6. AGLOMERACIJA

- ciljen:

- povećanje zrnatosti - ujednačavanje troškova komunikacije i računanja
- prilagodljivost uo druge uvjete rada (broj proc., vel. problema, ...)
- snajanje troškova implementacije

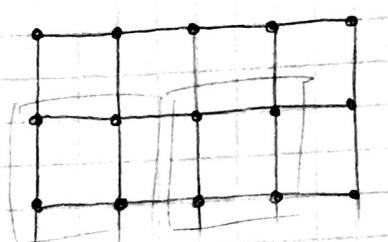
### 5.6.1. POVEĆANJE ZRНАTOSTИ $\rightarrow$ najvećije

- snajiti vel. poruka ili broj poruka koje se šalju u programu

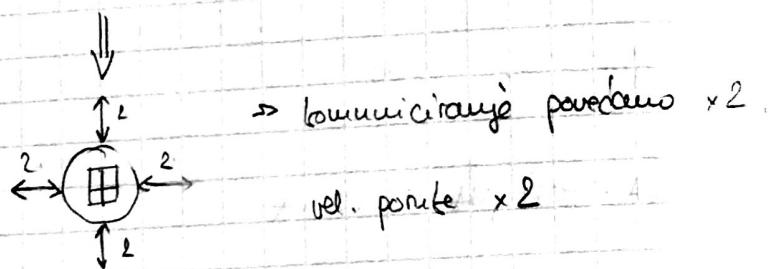
① POVEĆAVANJE ZADATAKA = grupiranje više zadataka u jedan (najbolje po svim dimenzijama strukture zadataka)

- lanc , 2D, 3D prostor ...

QF.

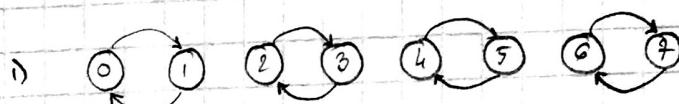


$\Rightarrow$  = 4 poruke prima i šalje  
(vel. 1 element)

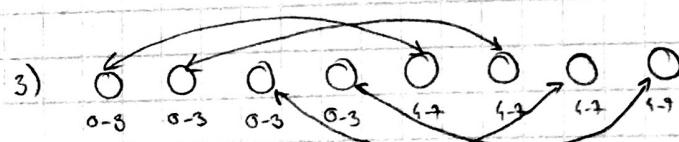
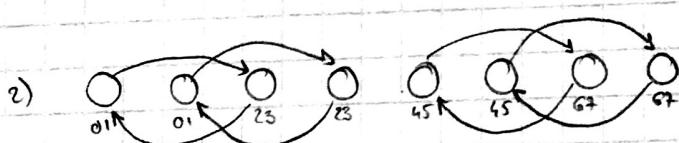


② UNIŠESTVUĆAVANJE RАČУНАЊА = povećanje kol. računanja s ciljem snajenja

- pr. sudi se uo all-reduce

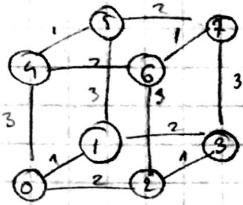


$\Rightarrow$  sudi operacije uo sekciju i delivrem



$\Rightarrow$  LEPTIR KOMUNIKCIJSKA STRUKTURA (BUTTERFLY)

## - HIPERKOCKA (HYPERCUBE)



## - OBRAZAC HIPERKOCKE

```
za ( i=0 ; i < log n ; i++ )
```

```
dest-ID = proces-ID XOR  $2^i$ 
```

```
send (mog-podatak, dest-ID);
```

```
recv (drugi-podatak, dest-ID);
```

```
operacije();
```

## 5.6.2. OCUVANJE PRILAGODLJIVOSTI

- broj zadataka: orisan o veličini problema, veći od broja procesora

- aglomeracija pozajmjuje svojstvo:

- podjednaka složenost zadataka
- želimo očuvati prilagodljivost broja zadataka
- što manji broj zadataka
- manji trošak pretvorbe u paralelni oblik

## 5.7. PRIDRUŽIVANJE

- koji procesor izvodi koji zadatak
- zajedno stvaraju pristup: jednoliko raspodjela strukture zadataka
- metode:
  - vjeđenjačavanje opterećenja
  - raspoređivanje zadataka

### 5.7.1. UJEDNATČAVANJE OPTERECENJA

- pretpostavka: stalan broj zadataka
- može biti:
  - STATIČKO: zad. se rasporedjuju ne poč rada
  - DINAMIČKO: vjeđenjačavanje se potrebuje više puta
    - DINAMIČKO - lokalno (gledeći se susjedi, užeti/dati zad. susjedu)
    - globalno (totalni reset, svi su se raspodijeli iz početka)
- REKURZIUNI BISEKCIJA = podjela strukture zadataka
  - podjela na temelju komunikacijskih i/ili racionalnih troškova
- LOKALNI ALGORITMI = pouzatije usporedbu susjednih procesora
  - mali trošak (relativno jednostavno se vidi da li je hešo?)
  - relativno spore primjerkeba ve stotinu promjena
- VJEROJATNOSNE METODE = slučajne podjela pod. po procesoru
  - mali trošakovi
  - radi unije
  - mogući veliki troškovi komunikacije (ako su u nesmetku slomili)
- CIKLICKO PRIDRUŽIVANJE = uz p procesora jedan procesor dobiva svaki p-ki zadat

pr.	P0	21	24	27
	P1	22	25	28
	P2	23	26	
zadatak				

### 5.7.2. RASPOREDJIVANJE ZADATAKA

- pretpostavka: više zad. trajući životnuog vijekta
- voditelj / rodnik: voditelj dodjeljuje zad. radniku
  - potencijalno zagrijevaju voditelja
- hirerarhijski voditelj / rodnik: postoji podgrupe roditelje s dcl. voditeljima
- decentralizirane metode: ne postoji zajednički skup zadataka (zad. jednoliko podjeljeni po rodućima)
  - potreban mehanizam otkrivanja završetka - crisi o modelu
  - pribružavanje: pozajmice suradnja
    - voditelj nije preopterećen (otkidi uzmajed ili ukuadaus)
    - isplativi trošak ujednačavanja opterećenja (kad već rođimo, to uvesto koštak → treba biti manji od planirane vrednosti)
- \* otvarani proučjiti za određeno pribružavanje

**ZAD.** Otkrivanje redoslijeda za ulazak u t.o. (MI 2015.)

- u procesu, svaki imao id:  $0 \dots n-1$ , svaki imao rbr ( $\in \{0, N-1\}$ ) ulazak u t.o.
  - odrediti redoslijed prethodnike i sljedbenika
- logaritamska složenost po broju poslanih poruka

\* hiperkocka

brojeni [ $n$ ] =  $\{N, \dots, N\}$ ; → ideja popuniti u svim procesima uz bazu strukt hiperkocke

brojeni [rbr] = 1D;

za ( $i=0; i < \log N; i++$ )

dest = id XOR  $2^i$ ;

send (brojeni [], dest) \* šalje poruku

receive ( $n$ -brojeni [], dest)

za (svaki elem j u brojeni [])

ako ( $n$ -brojeni[j] < N)

brojeni[j] =  $n$ -brojeni[j]

prethodnik = brojeni [rbr-1]

sljedbenik = brojeni [rbr+1]

\* HPI → mogućnost zastoje

\* ovise o implementaciji

\* **nebitno** za zadatku

pr. blagajna - kva

→ PODJELA - funkc. dekompozicija - blagajne jednoliko dijele zaujme

→ KOMUNIKACIJA - 3 vrste poruka: → asinkrono

① zahtjev

② odgovor

③ info - informacije o sjedalima tako se tko kaže moći blagajnici

→ AGLOMERACIJA - jedna blagajna odgovara jednom procesu

→ PRIDRUŽIVANJE

\* usta blagajne ustan zaušeta može probati još koje sjedalo

↳ ve može se sprijeciti da ustan što jedan zauš ostali ne uspije

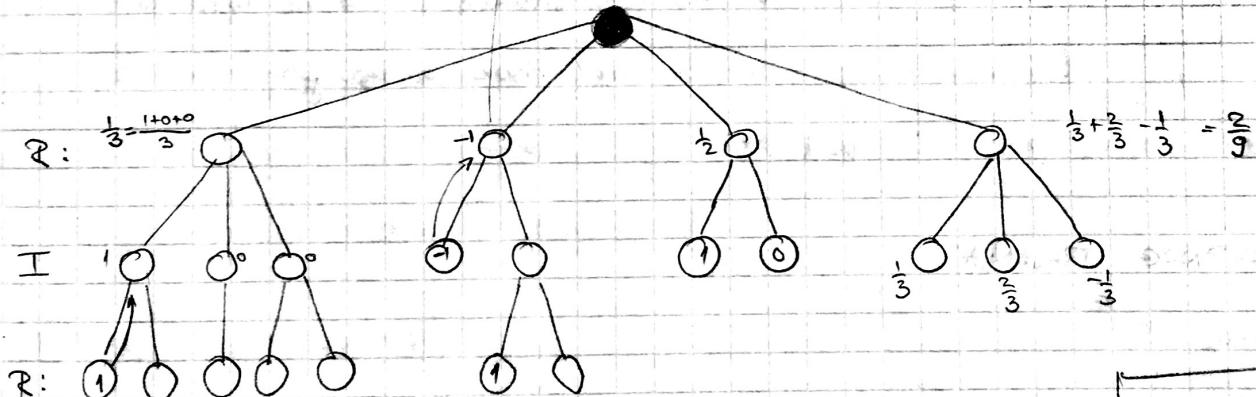
### 3.9. PRIMJER:

DZ 2 | 4 U NIZU

neči odigrati je  
možni izgubiti

- ili dat. ili zauđe

-1 = igrač pobjedi  
1 = comp. pobjedi



\* br. procesora: 1-8

#P	2	3	4	5	6	7	8
t	4	3	2	2	2	1	1
	322	2221					

} NE  $\Rightarrow$  više redukta  
(zauš za red vrvi)

veću podjelu  
jer veću više redukta

\* rez blagajne barem 3  
(velika gomila u s)

## 6. KVANTITATIVNA ANALIZA PARALELNOG ALGORITMA

### 6.1. DEFINIRANJE PERFORMANSI

- osnovni kriteriji

- multimedija, kapacitet (procesor)

- sustav upravljanja, vrijeme odziva

- računari: učinkovitost

- proučavani kriteriji

- ukupno trajanje izvođenja neke paralelne aplikacije ( $T$ )

- stalabilnost programa

#### 6.1.1. AHDATHOV ZAKON

#### 6.1.2. OPTIMIZACIJA IZ PRIMJERA

- metode u zadanim okruženjima  $\rightarrow$  nepotpuna informacija

#### 6.1.3. ASIMPTOTSKA ANALIZA

$$T_1 = 1 \text{ proc.}$$

$$T_p = p \text{ procesora}$$

$$S = \frac{T_1}{T_p} \quad E = \frac{T_1}{p \cdot T_p}$$

$$S_{p=4} = 2$$

$E = 0.5 \Rightarrow$  pola vremena se trudi na nešto što nije korisno računajući

$$P = 1 \dots 8$$

#### 6.2. MODEL UKUPNOG TRAJANJA

- ukupno trajanje:  $T$

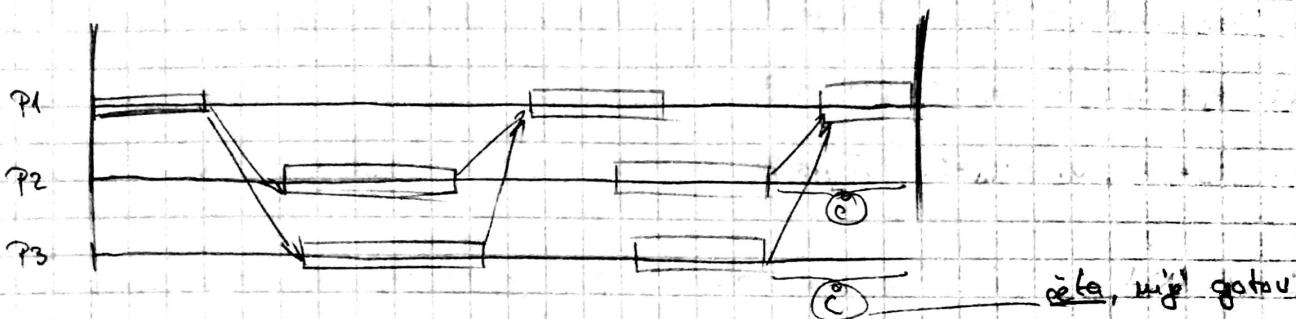
$T_i^r$  = trajanje računanja na  $i$ -tom procesoru

$T_k^i$  = trajanje komunikacije na  $i$ -tom procesoru

$T_c^i$  = trajanje čekanja na  $i$ -tom procesoru

$\Rightarrow$  trajanje izvođenja:  $T = \text{zbog trajanje računanja, komunikacije i čekanja}$

\* ukupno trajanje većini o procesoru



↳ početek bilo koji proces počinje

↳ kraj = bilo koji proces završi

a) TRAJANJE NA i-tOM PROCESORU

$$T = T_R^i + T_k^i + T_C^i$$

b) PROSJEK NA SVIM

$$T = \frac{T_R + T_k + T_C}{3}$$

### 6.2.1. TRAJANJE RAČUNANJA ( $T_R$ )

=  $T_R$  = ukupno trajanje računanja većini procesora

- određivanje:

a) kao sljedui program (veću uvođenju računanja)

b) funkcije veličine problema broja zadataka i načina raspodjele po procesoru

(že svaki pojedinačni skup je računa)

### TRAJANJE KOMUNIKACIJE

=  $T_k$  = trajanje komunikacije većini procesora

- ocjenjuje se trajanje slanja poruke (broj kom. se podeljuje proc. koji šalje)

\* mijene primaju u  $T_C$

- 1 poruka: zbroj trajanja postavljanja i trajanja slanja svih nječi poruke

$$T_{msg} = t_s + \frac{tw \cdot L}{\text{post. sreću}}$$

prije: uva zavode:

- STM:  $t_s = 1.5 \mu s$ ,  $t_w = 0.00013 \mu s/B$
- SOKI:  $t_s = 13 \mu s$ ,  $t_w = 0.0011 \mu s/B$
- LAN:  $t_s = 100 \mu s$ ,  $t_w = 0.0095 \mu s/B$

TRAJANJE ČEKANJA: - sve što je preostalo; ugl. zavremensko

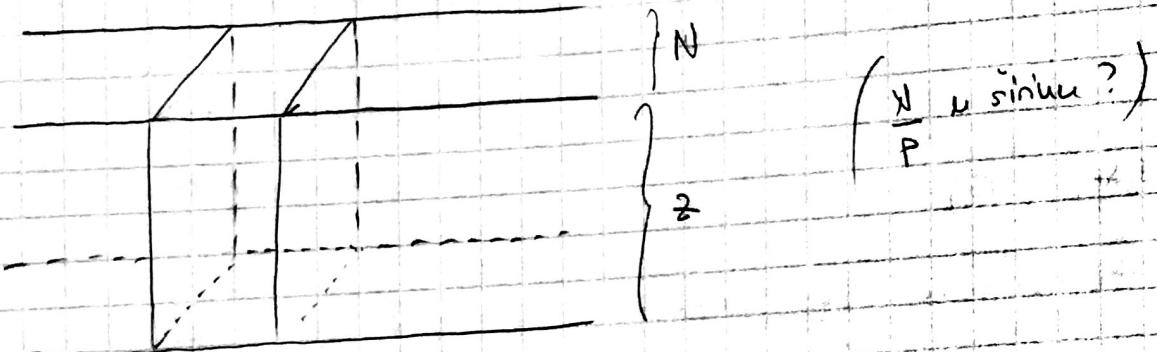
uzroci:

- vema posta
- ima posta, ali nema podatka

pr Trajanje izvođenja akomstenskog modela

① ID

- \* grupiramo i u dubinu



- trajanje:

$t_c$  = trajanje računanje jedne točke

$$T_e^i = t_c \cdot Z \cdot N \cdot \frac{N}{P} = t_c \frac{N^2}{P} \cdot Z$$

- KOMUNIKACIJA (uz  $\frac{N}{P} \geq 2$ )

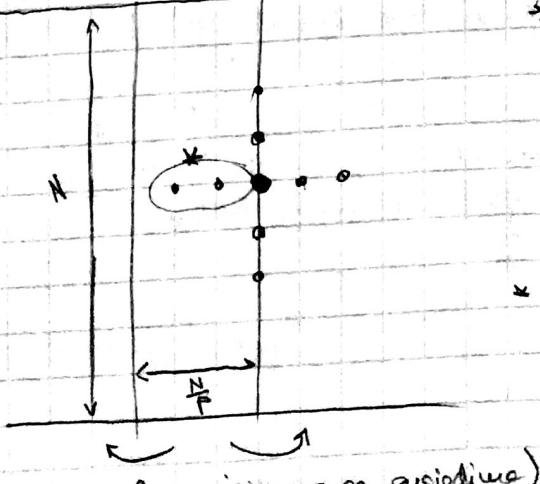
- floot →

→ varijabli da je svaki god = 2 reda proc.

↳ komuniciraju sam se susjedima, te te se

oni u dalje

\* daju 2 reda susjedu i on meni



- Šaljemo 2 poruke

- 2 ribna polja su  $2 \cdot N \cdot 2$  podataka

→ komunikacija:

$$T_k = 2 \cdot (t_s + t_w \cdot 2 \cdot N \cdot 2)$$

✓  
2 poruke

10. punkt

[21]

\* jedan od mnogih zad  
na kojima će se bazirati  
zad uo 21

- ŠEKTANJE

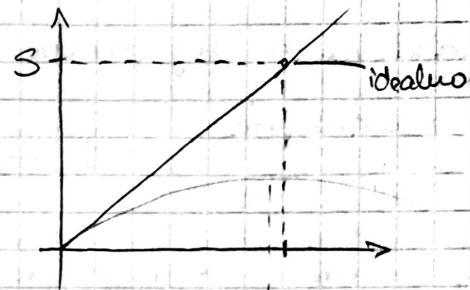
- identični procesori → isto se racuna → veoma šekanje

$$\Rightarrow T = t_c \cdot \frac{N^2}{P} \cdot 2 + 2t_s + 4t_w \cdot N \cdot 2 \rightarrow \text{trajanje } \xrightarrow{\text{1 iteracija}}$$

## 6.2.2. UBRZANJE I UČINKOVITOST ALGORITMA

UBRZANJE:

$$S = \frac{T_i}{T_p}$$

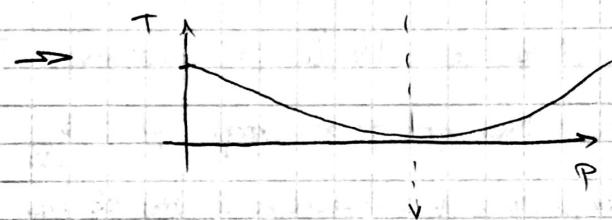


UČINKOVITOST:  $E = \frac{T_i}{P \cdot T_p} = \frac{S}{P}$

\* u konji ograničeno, u praksi bolje

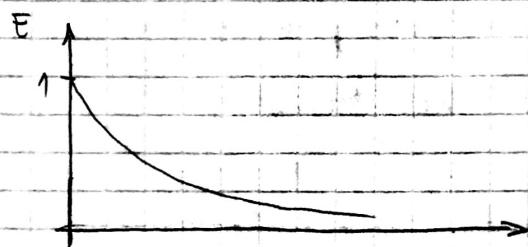
\* ubrzanje monotono pada,

ali može i rasti



← moguća situacija

gori max odgovore  
dovezen min



⇒ učinkovitost unijek monotono pada

60% uk. trajanja ⇒ 60% vremena se troši na

prečenje, a 40% kamenje

čekanje

PF: 1D atm. model

$$E = \frac{T_1}{P \cdot T_P} = \frac{t_c \cdot N^2 \cdot 2}{t_c \cdot N^2 + 2 \cdot t_s \cdot P + k t_w \cdot N \cdot 2 \cdot P}$$

overhead  $\Rightarrow$  komunikacija

$\hookrightarrow$  RELATIVNA učinkovitost i utrošak  
(ako veće vise)

\* RELATIVNO = u odnosu na isti algoritam tje. program

\* ABSOLUTNO = u odnosu na najbolji sljedeci algoritam

$\hookleftarrow$  inspiracija za puno zad.

\* Z1

### 6.3. ANALIZA SKALABILNOSTI

-2 sručnje:

#### a) PONŠANJE UZ STAVNI PROBLEM

- tako se prog. poveća kada veći broj procesora

- učinkovitost: monotono pada

- trajanje: može i rasti i padati

- ako povećamo broj procesora : trajanje programa , koliko ujutru procesora učini u minimum učinkovitost

[Zad] Paralelni rač. plaća se 1 kn/lu po procesoru. Trajanje sljedećeg prog. je 10 dana

a učinkovitost dostupnog paral. prog. je  $E = \frac{3}{2+P}$ . Na raspolaganju je 500 kn

koje je min trajanje koje možemo platiti tek ne koliko proc.?

$$C = P \cdot T_P$$

$$T_1 = 240 \text{ lu} \quad (\text{trajanje sljed. alg.})$$

$$C_{\max} = 500 \text{ kn}$$

$$\frac{T_1}{P \cdot T_P} \rightarrow C$$

$$C_{\max} = \frac{T_1}{E_{\min}} \Rightarrow E_{\min} = 0.48$$

$$E_{\min} = \frac{3}{2+P_{\max}}$$

$$P_{\max} = 6.25$$

$\hookrightarrow$  možemo ga primiti

$$\rightarrow C = \frac{T_1}{E_{\min}} = 480 \text{ kn} \quad T_P = \frac{T_1}{P \cdot E_P} = \frac{C}{P} = 120 \text{ lu} \quad \text{upr. } [4.75] \Rightarrow 4$$

- koliko je vremena trajanje projekta uz pravovoljni br. proces?

$$T_p = \frac{T_1}{P \cdot E_p} + \frac{T_1 \cdot (2+p)}{3 \cdot p} = \frac{2T_1}{3p} + \frac{T_1}{3}$$

27.5.2019.

zad 12.30

za spust

$$\lim_{t \rightarrow \infty} T_p = \frac{T_1}{3} = 80 \text{ dnevi } \approx 3 \text{ mesecev}$$

### b) PONASANJE UZ PROMJENJIVU VREDNU PROBLEMA

- pitanje

~~$$T_p = N + \frac{N^2}{P}, T_1 = N^2, C_p = P \cdot T_p$$~~

a)  $N=10 \Rightarrow T_1 = 100, C_{max} = 200 \$$

$$T_p = 10 + \frac{100}{P}, C = P \cdot T_p = 10P + 100 = 200 \Rightarrow P=10$$

$$T_p = 10 + \frac{100}{10} \cdot 20, S = \frac{T_1}{T_p} = 5, E = \frac{3}{P} = \frac{3}{10} = 0.3 = 30\%$$

↳ uverz. je 5  
na 10 proc

b)  $N=20 \Rightarrow T_1 = 400, C_{max} = 800 \$$

$$T_p = 20 + \frac{400}{5}, C = 20P + 400 = 800 \Rightarrow P=20$$

$$T_p = 20 + \frac{400}{20} = 40, S = \frac{T_1}{T_p} = \frac{400}{40} = 10, E = 0.5 = 50\% \quad \hookrightarrow \text{ispod toga je ne isplati}$$

a scenarij  $\Rightarrow$  b scenarij  $\Rightarrow$  kol. poslo :  $T_1 = 100 \Rightarrow T_1 = 400 \Rightarrow \times 4$

broj procesora:  $N=10 \Rightarrow N=20 \Rightarrow \times 2$

↳ ako se poveča kol. poslo, broj proc se poveča  $\Rightarrow$  kol. poslo kota bi učinkovitost ostale iste = IZOUČINKOVITOST  $\rightarrow$  prenjeva: zato je  $\Rightarrow$  n

$$E = \frac{T_1}{P \cdot T_p} = \frac{N^2}{N_p + N^2} \Rightarrow N^2 = E \cdot (N_p \cdot N^2)$$

$$N^2 \sim E \cdot (N_p \cdot N^2) \quad (N \text{ se menjajo } \Rightarrow \text{kako se ovisi } \Rightarrow \text{ menjajo da je } E \text{ konst.})$$

uvjetimo  $\rightarrow$

$$N = f(p) = ?$$

$N^2$  je prop. kol. poslo,

$N$  nije

$$N = \sqrt{P}, \text{ (P) } P^2, \dots$$

$$N = P \Rightarrow N^2 \sim (NP + N^2)$$

$$P^2 N \sim (P^2 + P^2) / P^2$$

$\rightarrow$  konst (ne ovisio ničemu više)

$$\Rightarrow N \sim O(P)$$

$\hookrightarrow$  mora vrijedati da bi učinak ostala ista

IZOUČINKOVITOST  $\rightarrow$  kako se kol. posta mora mijenjati da učinak ostane isti  
 $(T_1)$   
 $= O(P^2)$

$T_1, P, E$

$P' > P \Rightarrow$  učinkovitost pada

\* da bi učinkovitost ostala ista tražimo  $T' = f(P) = ?$  ( $E = \text{konst}$ )

IZOUČINKOVITOST = promjena vel. posta u ovisnosti o promjeni broja procesora  
 stalnu učinkovitost (može i obrnuto : promjene # proc.

- PRIKAZ : funkcija br. procesora u 0 motociji

1D akut. model



$$T_{1D} = t_c \frac{N^2}{P} Z + 2t_s + 4t_w \cdot N \cdot Z$$

$$E_{1D} = \frac{t_c N^2 Z}{t_c N^2 Z + 2t_s P + 4 \cdot t_w \cdot N \cdot Z P}$$

$$t_c N^2 Z \sim E \cdot (t_c N^2 Z + 2t_s P + 4t_w \cdot N \cdot Z P) \rightarrow \text{zeliemo da vremo u } N \text{ u } P,$$

$$N = f(P) = P$$

da se sve zakrabi

$$t_c P^2 Z \sim E \cdot (t_c P^2 Z + 2t_s P + 4t_w \cdot P^2 Z)$$

$$t_c Z \sim E \left( t_c Z + 2 \frac{t_s}{P} + 4t_w \cdot Z \right) \Rightarrow \text{OK za br. proc. koji nije pravil}$$

$$\text{POSTO } (T_{1D}) \sim N^2 \sim P^2$$

$$\Rightarrow \text{izoučinkovitost} = O(P^2)$$

$\hookrightarrow$  ako povećamo broj proc. 10 puta, kol. posta povećamo za 100

- gledaju

①  $N = P$

②  $N = C$

ali mi

pr. 2  $\Rightarrow$  ah

- jedan

$\rightarrow$  u

- 200

$\leftarrow$

- gledajući svaki od 3 dimenzija zasebno:

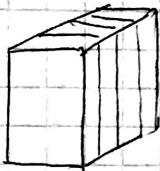
$$\textcircled{1} \quad N = P \quad t_c P^2 \leq N \leq (t_c P)^2$$

$$\textcircled{2} \quad N = \sqrt{P}$$

ali mi učinimo nejgoni slúčaj

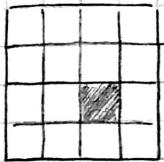
$\Rightarrow$  Na 21 se sve more počekati

pr. 2  $\Rightarrow$  akt. model



$\Rightarrow$  jednake širine i dubine viseća?

- jedan zad. u flochu:

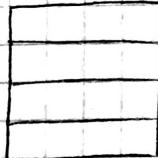


N

N

P zadatka

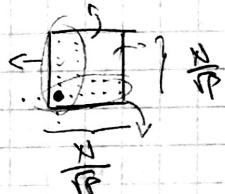
Pre:



(vsiče)

$\Rightarrow$  u jednoj dimenziji  $\sqrt{P}$  zad., u drugoj  $\sqrt{P}$

- zad.: (\*unijek visok  $\geq 2$ \*)



$$\left(\frac{N}{\sqrt{P}}\right) z^2 = \frac{N^2 z^2}{P}$$

morame znati da  
magri izračunati  
svaku stranicu

$\Rightarrow$  moram barem 2 sloja točaka

$\Rightarrow$  komunicira samo s neposrednim susjedima

- RACUNATJE:  $T_R = \left(\frac{N}{\sqrt{P}}\right)^2 z \cdot t_c = t_c \frac{N^2}{P} \cdot z$

- KOMUNIKACIJA:  $T_k = k \cdot (t_s + 2 \cdot \frac{N}{\sqrt{P}} z)$   
(k poruka)

$$\frac{N}{\sqrt{P}} \cdot z \cdot 2 \Rightarrow \text{tol. pod.}$$

ostaci zad. u 21

$$\Rightarrow T_{2D} = t_c \cdot \frac{N^2}{P} \cdot z + k \cdot t_s + 8 \cdot \frac{N}{\sqrt{P}} \cdot z$$

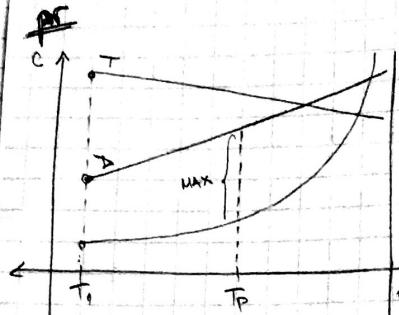
$$E = \frac{t_c N^2 z}{t_c N^2 z + k t_s P + 8 N z \sqrt{P}}$$

$$N = f(P) = \sqrt{P} \Rightarrow E = \frac{t_c P^2 z}{t_c P^2 z + k t_s P + 8 z P}$$

posao:  $N^2 N P$   
uncinkost  $= O(P)$

- bolje od  $T$
- najbolje moguce (ne moze biti od linearne)

$\Rightarrow$  optimizacija linearne izostanakost



Cijena

$$a) D = \text{dubit} \quad (\text{oni} \text{ i o trenutku})$$

$$z = \text{zarađa} = D - C$$

$$b) TR = \text{trošak} : \min(C + TR)$$

230 (21)

Parallelno računalo pomoći  $\approx 1 \text{ h} / \text{l} \text{ po procesoru}$ ,  $c$  paralelno program uvećava izvodenje  $T_P = 50 + \frac{150}{P} \text{ h}$ . Dobit od rez. programa oni i o trenutku dobivajući opisano je izrazom:  $D = \max(0, 18(T_1 - T_P))$ . Kako izračunati izvodenje čime najvećim mogućim zarađu ( $D - c$ ) i ne koliko proc?

$$z = D - c = 18T_1 - 18T_P - \frac{P \cdot T_P}{c}$$

$$\begin{aligned} T_1 &= 50 + \frac{150}{1} = 200 \\ &\quad \xrightarrow{\quad} 18 \cdot 200 - 18 \cdot 50 - \frac{18 \cdot 150}{P} - 50P - 150 \\ &= \dots - \frac{2900}{P} - 50P \end{aligned}$$

$T_P = 7.6$
$c = 500$
$D = 2314$
$z = 18014$

$$\frac{2900}{P^2} - 50 = 0$$

$$P^2 = 54$$

$$P = \sqrt{7.348} \Rightarrow P = 7$$

$$\cancel{\Rightarrow \text{kor. } P = 7.68 \Rightarrow P = 8}$$

## 6.4. PROVERA

- USPOREDA
- uzroci mogle
- model
- program
- rezultat
- moguci rješenja
- nejednačine
- unesene
- vrednosti
- pogreške

pr.  $T_1$

$T_P$

$T_{kP}$

~~230~~ atm.  $w$

$\rightarrow$  sudi  $z$

### 6.4.1. NEFT

- ANOMALIJE

$w$

$s$

## 6.4. TROJERNI MODEL I IMPLEMENTACIJA

### - USPOREDBA STRUKTURA I PREDVISENOG RONITANJA

#### - uroči maliči:

- model uverćem ili nepravilno  $\rightarrow$  krajnje

- program nije u skladu s modelom

- rezultati ili metoda uverćeg mogu biti nepravilni

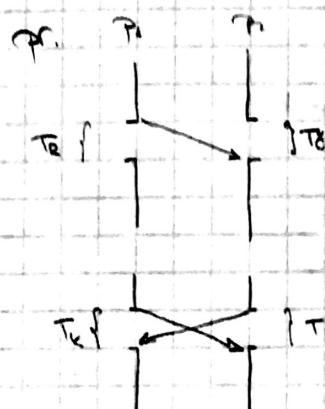
#### - mogući nedostaci modela:

- nejednakost opterećenja u radu

- usisnuće rezerviranje

- nestabilni algoritmi i plata

- ograničeni kapacitet komunikacije



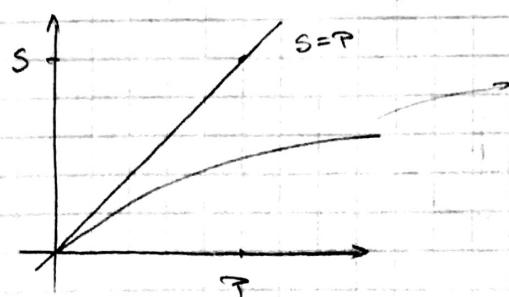
$\rightarrow$  ovisno o tehnologiji (npr. kod MPI-ja možda ne protezi)

#### ath. model

$$\rightarrow \text{sudj. zod. racuna: } T_k' = t_c \cdot \frac{N^2}{P} \cdot 2 = t_c \left[ \frac{N}{P} \right] N \cdot 2$$

### 6.4.1. NEPRAVILNOSTI UBRZANJA ALGORITMA

- ANOMALIJA UBRZANJA = ako program traje kraće od predviđenog, a rezultat je veća za veći  $\Rightarrow$



- ubrzanje veće od broja procesora je SUPERLINEARNO

#### UBRZANJE

#### - mogući uroči:

- pričuća memorija (cache, virt. memorije)

- anomalijske pretraživanja

- mogući učinci:

- povećana memorija (cache, virtualna memorija)

- manjši vremenski razdoblje

→ pr. binarni prepoznavanje

| X |

1P: ↑

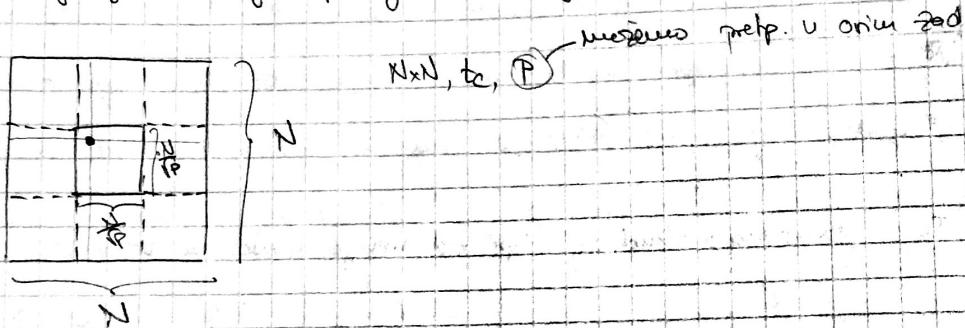
2P: ↑      ↑

ubrajuje:  $\sim N^2$

ali ovisi gdje se nalazi

2AD

Paralelni program kvadrira matricu  $N \times N$  tako da je matrica podijeljena u zad. kvadratnog oblika jednake veličine. Trosak množenja je to. Koji je trajanje izvođenja operacije kvadriranja. Izvoričnost?



$\frac{N^2}{P}$  točaka

$$\Rightarrow \text{RAČUNANJE: } T_R = \frac{N^2}{P} \cdot N \cdot tc = \boxed{\frac{tc \cdot N^3}{P}}$$

broj množenja

⇒ KOMUNIKACIJA:

(svim ostalim zad. u  
redu i stupcu)

$2(\sqrt{P}-1)$  ponuka s  $\frac{N^2}{P}$  podataka

#zad-ja  
koris  
i vert.

$$T_K = \boxed{2(\sqrt{P}-1)(t_s + t_w \cdot \frac{N^2}{P})}$$

$$\Rightarrow T = tc \frac{N^3}{P} + 2 \cdot (\sqrt{P}-1) \left( t_s + t_w \cdot \frac{N^2}{P} \right)$$

- UČINKOVIT

- IZVUČIN

tc

- POSAO

⇒

TZ: is

- VČINKOVITOST:  $E \propto$

$$\frac{t_c N^3 + 2(t_p - 1)(t_o P + t_w \cdot N^2)}{N \sqrt{P}}$$

zvijezda

- IZVČINKOVITOST:  $N = \sqrt{P}$

$$t_c \cdot \sqrt{P} = E \cdot (t_c \cdot P^{\frac{3}{2}} \cdot 2 \cancel{t_p^2} + 2 t_w P^{\frac{1}{2}})$$

- POSTOJ  $\sim N^3 \sim P^{\frac{3}{2}}$

$$\Rightarrow \text{IZVČINKOVITOST: } \boxed{O(P^{\frac{3}{2}})}$$

D2: isti zad., ali je matrica podijeljena na stupce  $\rightarrow$  1D podjela

## 7. RAZVOJ MODULARNIH PARALELNIH PROGRAMA

- par. program → modul → kompozicija
- moduli nude fiksne ili dinamičke podjelu podataka
- različiti moduli mogu zahajevati pre rasподјelu podataka

### 7.2. NACINI KOMPOZICIJE MODULA

#### 7.2.1. SLIJEĐENA KOMPOZICIJA

- moduli se izvode kao slijed (jedan za drugim)
- prednosti:
  - jednostavnost, čitljivost
  - neva komunikacije između modula
- nedostaci:
  - nepromogodljivi broj procesora

#### 7.2.2. PARALELNA KOMPOZICIJA

- različiti moduli izvode se istovremeno na različitim procesorima
- prednosti:
  - bolja raspodjela operacija (koliko proc. koliko)
- nedostaci:
  - povećani trošak komunikacije

#### 7.2.3. ZAJEDNIČKA KOMPOZICIJA (nosi MPI model to ne podržava)

- različiti moduli izvode se istovremeno na istim procesorima
- prednosti:
  - podeljivanje zad. je nevezano o ostalim modulima
- nedostaci:
  - izmjenjuju kontekste
  - veći trošak memorije (veći broj se isti modul mora imati pobrani)

### 7.3. MPI I MODULARNO PROGRAMIRANJE

- uz pomoć komunikatora = grupe procesa  $\rightarrow$  izvode 1 modul  
(MPI-COMM-WORLD)

- 2 operacije: kopiranje i dijeljenje komunikatora

#### 7.3.1. KOPIRANJE KOMUNIKATORA

- stvara se istovjetna kopija postojećeg komunikatora
- novi komunikator je novi kontekst komunikacije: poruke je moguće zauzimati  
(parallelnom modulu dodjeljuje se posebni komunikator)
- omogućuje sledeću kompoziciju:

#### 7.3.2. DIJELJENJE KOMUNIKATORA

- jedna grupa dijeli se u više podgrupe
- pripadnost istoj podgrupi definira se jednakom vrijednošću parametra color
- omogućuje parallelnu kompoziciju (više modula s više kom. na razm. proc.)

#### 7.3.3. STVARANJE MEĐUKOMUNIKATORA

- samo point-to-point komunikacija procesa iz različitih grupa

#### 7.3.4. OSTALE FUNKCIJE

## 8. PARALELNI EVOLUCIJSKI ALGORITMI

\* Gde boka funkcije

- EA : GA, GP, ES, ...

- podjela PEA:

- TPEA (trivijalan)

- RPEA (raspodjeljeni)

- MPEA (mosaici)

- GPEA (globalni)

- HPEA (hierarhijski)

### ① TPEA (TRIVIJALAN PEA)

- izvodi se istog evol alg. na više procesora neovisno

### ② REA (RASPODJELENI EA, DGA, ISU, GA)

- populacija se dijeli u više subpopulacije koje evoluiraju neovisno

a) MIGRACIJSKI INTERVAL = koliko često se obavljaju migracije

- kompletan (sudjeli 5)

- primjerice (stvaraju, mijenjaju)

- migracija može biti

- sinkrono

- asinkrono - posebno u sastoj populaciji

#### b) MIGRACIJSKA STOPTA

- broj jedinika koje se šalju

#### c) ODABIR ~~ZADATAKA~~ ~~ELIMINACIJE~~ BOGLJI JEDINKI

- upr. najbolje slučajne, odabranje selektivno

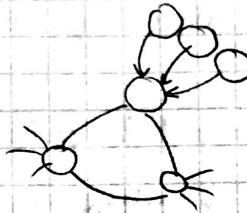
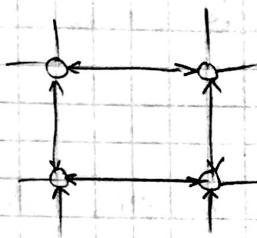
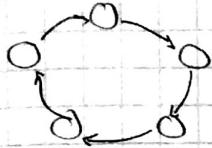
#### d) ODABIR ZADATAKA ZA ~~ELIMINACIJU~~ ELIMINACIJU

- koje jedinike ćemo eliminirati u lokalnoj subpopulaciji

- lošije jedinice

### e) TOPOLOGIJA MIGRACIJE (topologije razvijene jedinici)

→ prototip:



- OSTVARENJE (?)

- PODJELA: podatke

- KOMUNIKACIJA: lokalna

- AGLOMERACIJA: jedan zod., jedna subpopulacija

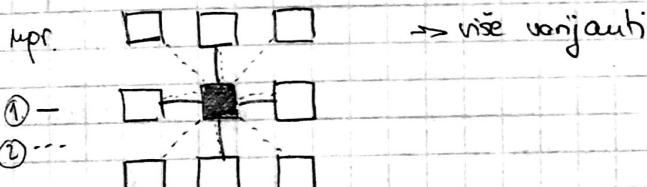
- PRIDRUŽIVANJE: jedan zod. na jednom procesoru

### ⑤ MPEA (MASOVNI PEA)

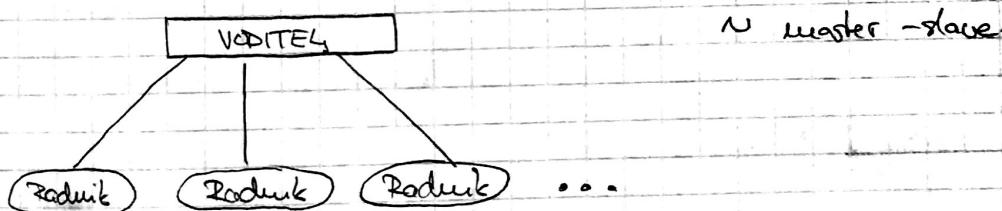
- granicni PEA

- jedan proc. radi nad jednom ili uvećanim skupom jedinici (superorganizam)

- SUSJEDSTVO = to je jedinice su u interakciji



### ⑥ GPEA (GLOBALNI PEA)



- zadizava se zajed. populacija (globalne), ali se dijeli tko što radi u tom populaciju

- podjela posta:

a) voditelj: podjela, selekcija

rodinci: evaluacija

b) voditelj - selekcija

radnici - evaluacija + genetske operacije (upr. mutacija)

c) voditelj - raspodjela posta

radnici - sve ostalo (selekcija, gen. oper., evaluacija)

- 2 varijante:

→ **SINKRONI** - isto iste svojstva kao i odgovarajući dijeli algoritam  
(poušće se identično neovisno kako suvremeno kada suvremeno podijeli zad.)

→ **ASINKRONI** - pristup podeljivim strukturama niz modusobno isključujući jedinice

## ⑤ HPEA (HIERARHIJSKI PEA)

- kombinacija više prethodnih modela

upr. REA + GREA

REA + RFA (?)

REA + HPEA (?)

- ostvaruje:

- podjela: funkcionalna

- komunikacija: globalna

- aglomeracija: 1 radnik - 1 zadatak

- približivanje: 1 zadatak - 1 procesor

- broj iteracija / generacija

→ gledamo isključivo broj evaluacija

- konvergencija: uvođene do zadatog rješenja (više puta)

- začinje kvalitete: potrebe se mijenjan najboljih  $\Rightarrow$  i svih podzadatka

može

ZAD.

Siubromi generacijobi globalni paralelni algoritam (GRGA), zvodi se u  $p$  procesa, rednuka je:  $r=p-1$ :

voditelj:

za (sve rednute)

posalji  $\frac{N}{r}$  jedinici;

za (sve rednute)

primi  $\frac{N}{r}$  jedinice;

rednik

primi, jedinice;

evaluiraj (ocjeni);

posalji jedinice;

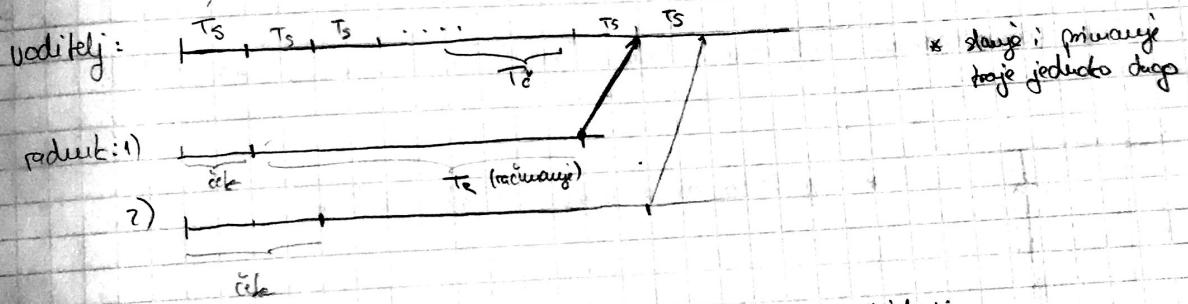
$$t_c \gg t_s, t_w$$

$\hookrightarrow$  trajanje evaluacije (ocjeni jedne jedinice

$$t_s, t_w$$

Obratite trajanje jedne iteracije algoritma

trajanje stanja  $\frac{N}{r}$  jedinici jedne rednute redniku



$$\hookrightarrow \text{pot} = r \cdot T_s$$

$$\text{sredina: } T_c \quad (\text{čekanje})$$

$$\text{kraj: } r \cdot T_s$$

$$T_s = t_s + t_w \cdot \left(\frac{N}{r}\right)$$

$$T_r = t_c \cdot \frac{N}{r}$$

$$(voditelj) \quad T = T_s \cdot r + T_c$$

$$S = \frac{t_c \cdot N}{t_c \cdot \frac{N}{r} + (r+1)(t_s + t_c \cdot \frac{N}{r})}$$

$$T_c = T_s + T_r - r \cdot T_s \quad (\text{čekao je samo ustan stanje})$$

$$S > 1 \Rightarrow \partial \text{ bi imalo gresku}$$

$$\Rightarrow T = 2rT_s + T_s + T_r - rT_s$$

$$= T_r + (r+1)T_s$$

$$= t_c \cdot \frac{N}{r} + (r+1)\left(t_s + t_w \cdot \frac{N}{r}\right)$$

$$\frac{\partial T}{\partial r} = \dots = \frac{t_c \cdot N}{r} + t_s + \frac{t_c \cdot N}{P} = 0$$

$$r = \sqrt{\frac{N(t_c + t_w)}{t_s}} \Rightarrow \text{broj radnika}$$

**ZAD.** izračunatost?

→ ne može se ostati u istoj razini učinkovitosti koliko god se poveća broj radnika

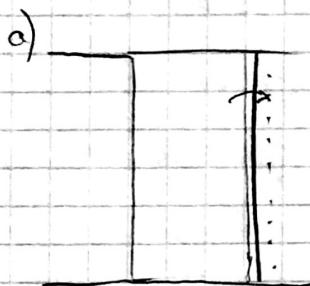
→ samo asymptotsko

(iz zvu)

**ZAD.** Parallelni algoritam računa elemente matrice tako da se može unijednost radnica uz pomoć unijednosti neprekidnih elem. gore i dole (moguće prebaciti počko ruba). Trošak računanja 1 elementa je  $t_c$ . Izrazite trošak  $T$  i terajući u  $P$  procesore te izračunatost ako je matrica podijeljena po:

a) stupcima

b) podmatricama  $\rightarrow$  izrač.: kolja (podjela po istom  $\sqcup$ )  $\Rightarrow O(p)$



\* Šefje desno da bi mogao usred izračunati

$$t_k = t_c \cdot \frac{N^2}{P}$$

$$t_k = t_s + t_w \cdot N \quad (\text{jedan početak desno u sredini})$$

$$\Rightarrow T = t_k + t_c = t_c \cdot \frac{N^2}{P} + t_s + t_w \cdot N$$

$$E = \frac{t_c \cdot N^2}{t_c \cdot N^2 + t_s \cdot P + t_w \cdot N \cdot P} \quad N \text{ konst.}$$

$N = \sqrt{P}$   $\Leftrightarrow$  opre

\* nazivnik ne smije rasti brže od novom (jednako je ok)

ocjena o 2

$$N = f(P) = P$$

$$\Rightarrow E \approx \frac{t_c}{t_c + t_s + t_w}$$

izračunatost:  $O(P^2)$