

PARALELNO PROGRAMIRANJE

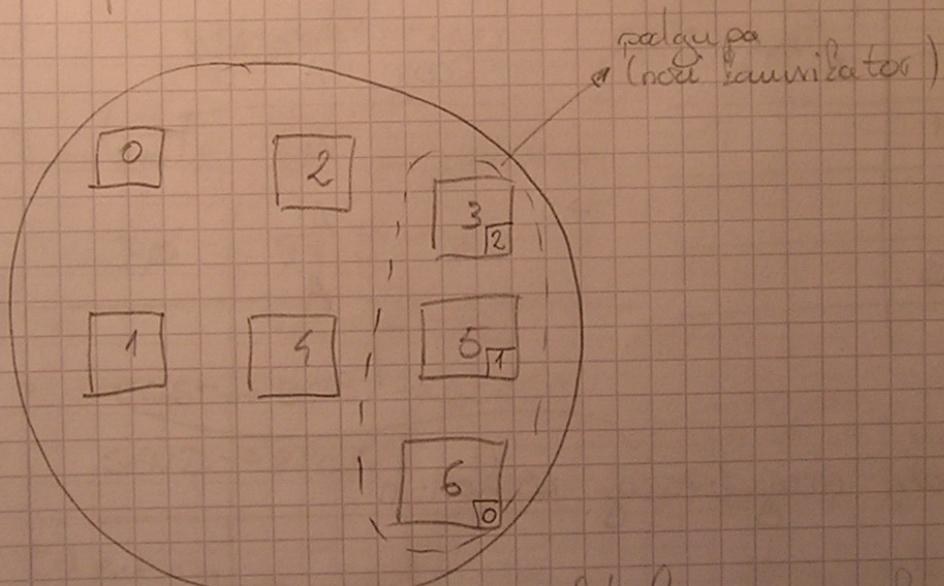
01.03.2010.

Računarske logike:

- ideja → posao se podjeli na jednake dijelove
- češ su postoje jekovi → O.K., ako nisu dobivamo istovrednost → broj dijelova poslova > broj procesora

MPI - ne nudi mogućnost konfiguriranja -> programer mora sami isprogramirati složni program - nudi samo mogućnost KOMUNIKACIJE!

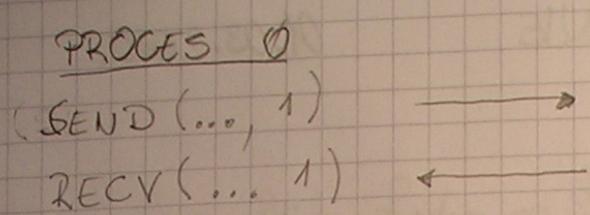
- program kod se pokrene mora utvrditi u koliko procesa je potrebno!



MPI_COMM_WORLD → globalna grupa logika činjena sve procese
→ korisnik može definisati podgrupe u kojima se obaveštava komunikacije

• Način funkcioniranja MPI:

- slanj i primanje : $\leftarrow \rightarrow$ (dušen.)

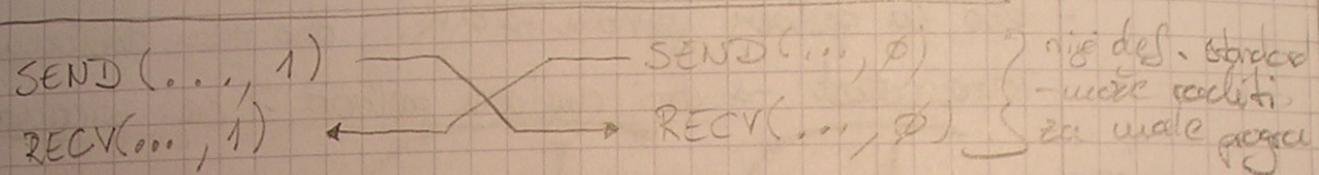


PROCES 1

```

RECV(..., 0)
SEND(..., 0)

```



- u kauflacionijim programima se radi rabljujaci send

GLOBALNA KOMUNIKACIJA

ROOT	A	
P1		
P2		
P3		



ROOT	A
	A
	A

Kad inicijalizacije se izvodi

Root	3	16
------	---	----

P1 9

P2 2

P3 7

SEND
BUF.

RECV.
BUF.

Zbroj se spaja

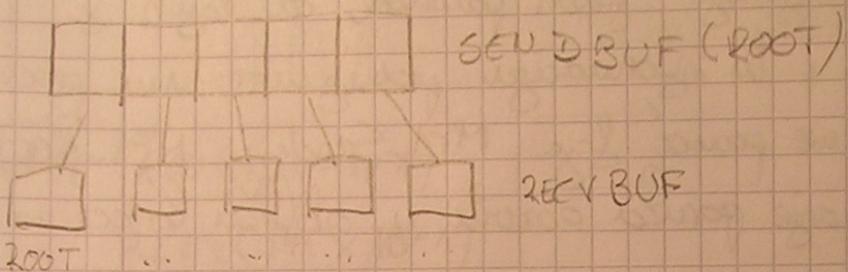
MPI-SUM
(sumiranje)

↳ CZ. se spaja na polovicu Root-a

08.03.2010.

P.P.

MPI - scatter - laisti se u pectoralni događaj



BROADCAST:

P \emptyset | P1

BCAST → BCAST
BCAST → BCAST

- uče i nauči se raditi

- ako je sve synchronizacija -
nije raditi!

P.3. Izrađe se 3 procesa

P \emptyset | P1 | P2

RECV ← SEND

a) BCAST → BCAST → BCAST

SEND → RECV

b) BCAST

SEND → RECV

→ BCAST

SEND

→ BCAST

{ da li će se dogoditi
a) ili b) nije doč.
standardsom.

Z1. MPI model:

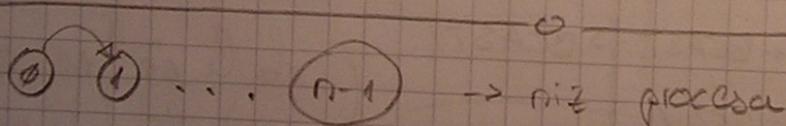
U M procesa

Svaki od N procesa posjeduje jedan podatak, od kojih je potrebno ažurirati najveću i tu informaciju poslati svim procesima. Napisati algoritam ve funkcije MPI_Send i MPI_Recv koji će slanje i primanje poruka obaviti u obliku koraka u 2 koraka.

- SKRAĆENA SINTAKSA MPI_Send : MPI_Recv
SEND (<ŠTO>, <KOME>)
RECV (<ŠTO>, <ODAKLE>)

- PODACI :

- n - procesa
- P - podatak
- ID - ID procesa
- MAX



- mogu se definisati 3 uloge

1^o 0 → polazni korak

2^o seri u skupini

3^o zadnji koji vrši rez.

AKO (ID == φ)

INACE AKO (ID == n-1)

INACE

AKO($ID = \emptyset$)

SEND(P, 1);
RECV(MAX, 1);

WAŻE AKO($ID = n-1$)

RECV(MAX, ID-1)
IF ($P > MAX$)
MAX = P;
SEND(MAX, ID-1)

WAŻE

RECV(MAX, ID-1)
IF ($P > MAX$)
MAX = P;
SEND(MAX, ID+1);
RECV(MAX, ID+1);
SEND(MAX, ID-1);

15.03.2010.

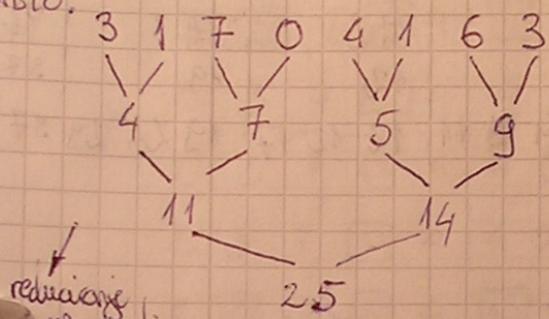
P.P.

ALGORITAM ZBROJA PREFIKSA

Pr. 1. +_PRESCAN NA PROIZVOLJNOM BROJU PROCESORA
- neognanice broj procesora

$$A[] = [3 \ 1 \ 7 \ 0 \ 4 \ 1 \ 6 \ 3], n=8$$

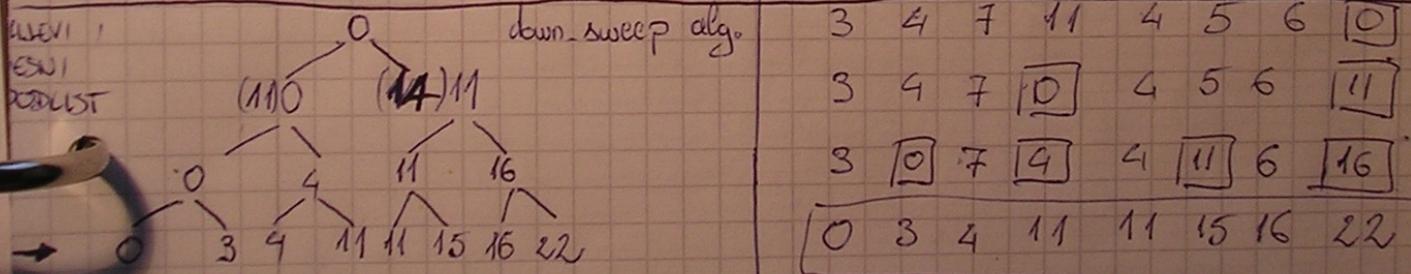
STABLO.



reducenje
u poziciju
broj procesora

MEMORIJA: (SHADED)

3	1	7	0	9	1	6	3
3	4	7	0	4	5	6	9
3	9	7	11	4	5	6	19
3	9	7	11	4	5	6	25

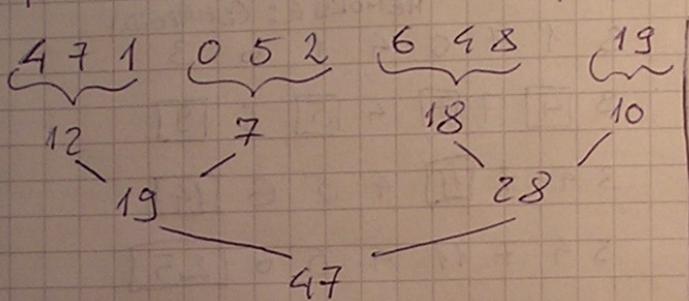


Pr. 2 + PRESCAN NA P PROCESORA
- ograničen broj procesora

$$A \rightarrow [] = [4 7 1 0 5 2 6 4 8 1 9], n=11, p=4$$

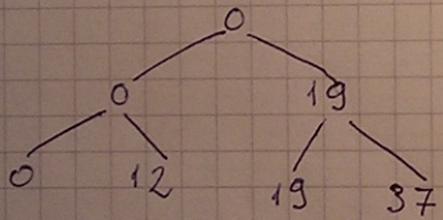
Svaki procesor zbroja "svogih" $\frac{n}{p}$ elemenata.

$$\frac{n}{p} = \frac{11}{4} = 2, \text{ ostatak } 3$$



P#1	P#2	P#3	P#4
0	12	19	37
11	12	25	29

DOWN-SWEEP



Pr.3 +_PRESCAN NA $n \neq 2^i$ (br. polja nije potencija broja 2)
- neograničen broj procesora

$$A[] = [2 3 7 1 4]; n=5$$

5 elemenata je isto kao i 8 elemenata - nadopunjimo:

$$\begin{array}{r} & \text{DODANO} \\ 2 & 3 & 7 & 1 & 4 & | & 0 & 0 & 0 \end{array}$$

$$\begin{array}{r} 2 & \boxed{5} & 7 & \boxed{8} & 4 & | & \boxed{9} & 0 & \boxed{0} \end{array}$$

$$\begin{array}{r} 2 & 5 & 7 & \boxed{13} & 4 & | & 4 & 0 & \boxed{4} \\ - & - & - & - & - & | & - & - & - \\ 2 & 5 & 7 & 13 & 4 & | & 4 & 0 & \boxed{0} \end{array}$$

$$\begin{array}{r} 2 & 5 & 7 & \boxed{0} & 4 & | & 4 & 0 & \boxed{13} \end{array}$$

$$\begin{array}{r} 2 & \boxed{0} & 7 & \boxed{5} & 4 & | & \boxed{13} & 0 & \boxed{17} \end{array}$$

$$\begin{array}{r} 0 & 2 & 5 & 12 & 13 & 17 & 17 & 17 \\ \hline & & & & & & & \end{array}$$

Pr.4. Rešiti Pr.3 ali za OGRAĐEN BROJ PROCESORA

Pr. 5. RADIX-SORT

NIZ: [5 7 3 1 4 2 7 2]

BITOVI: [1 1 1 1 0 0 1 0]

[4 2 2 5 7 3 1 7]

[0 1 1 0 1 1 0 1]

[4 5 1 2 2 7 3 7]

1 1 0 0 0 1 0 1

[1 2 2 3 4 5 7 7]

PARALELNA SORT OPERACIJA

5	7	3	1	4	2	7	2
1	1	1	1	0	0	1	0
0	0	0	0	0	0	2	2

PREDNO: 4 3 2 1 1 1 0 0
DUZINA:

GORE: 3 9 5 6 6 6 7 7

INDEX: 3 9 5 6 0 1 7 2

4 2 2 5 7 5 1 7

Pr. 6. CRCW PRAM - RAZRJEŠAVANJE ISTODOBNOG PISANJA

1° PRIORITETNO - u pisaju uspijeva samo procesor s najmanjim indeksom.

2° OPERACIJSKI - nad svim podacima koji se upisuju provodi se zadana operacija i upisuje rezultat.

3° SLUČAJNO - slučajno odabroni procesor uspijeva u pisaju (nederministički model)

! DEFAULT U ZADACIMA - MODEL 3° !

Primer modela (3^o):

- napisati alg. za CRCW PRAM rac. Tegi će za zadano polje P odrediti broj različitih davanata polja. Za polje od n davanata na raspodjelju je n procesora. Rezultat mora biti zapisan u 1 varijabli. Oceniti složnost algoritma.

$[1 \ 2 \ 1 \ 3 \ 4 \ 2 \ 5 \ 1] \Rightarrow 5$ (paraljizujući se ignoriraju)

PRAM.ALG.

REZULTAT = 1;

ZA i = 1 DO n-1 //akor prvič do prethodnog davanata

INC = 1;

PARALELNO (j = i + 1 DO n)

AKO (A[j] = A[i])

 | INC = 0;

REZULTAT += INC;

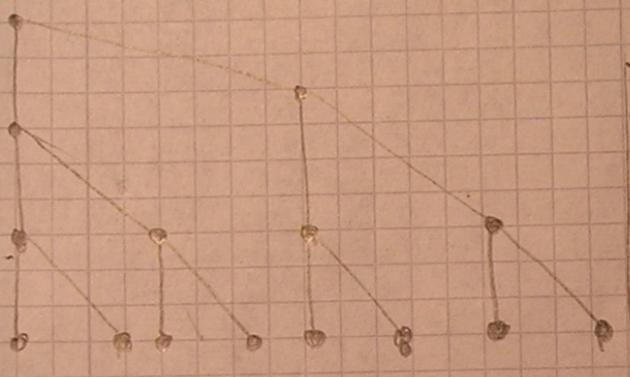
D.z. Napisaati alg. za EREW PRAM racunalo uz dostupnost procedure reduciranja za proizvoljnu binarnu operaciju

22.03.2010.

P.P.

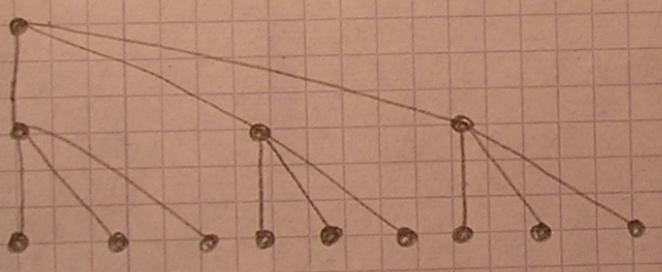
ASINKRONO PARALELNO RAČUNALO SA ZAJEDNIČKOM MEMORIJOM

a) BINARNO STABLO



PRAM	$\text{APRAM}_{d=3}$
{1}	{3}
{1}	{3}
{1}	{3}
{1}	{3}
$TR = 3$	$TR = 9$

b) TRINARNO STABLO

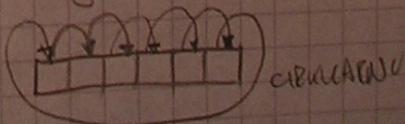


{2}	{4}
{2}	{4}
$TR = 4$	$TR = 8$

1. Napisati alg. za EREW PRAM rač. koji će za zadato polje A projektiti imena li u polju *duplicata*. Na raspodjeljivanju je prizvajanj broj procesora i procedura reduciranjia za prizvajaju binarnu operaciju.
Idea: svaki proc gleda svoj element polja sa sljedećim (ili svim) drugim, (tacim, ...)

PARALELNO ($p = 1 \dots n$)

$\boxed{\text{KOPISA}[P] = A[P]} \quad // \text{prizvano polje}$
 $\overline{\Sigma_A} \quad (d = 1 \dots n/2 + 1)$



CIRKULACIJA

PARALELNO ($p = 1 \text{ DO } n$)
 KOPUTA [p] = A [p]
 REZ [p] = \emptyset

$\frac{1}{n} A (d = 1 \text{ DO } n/2 + 1)$

PARALELNO ($p = 1 \text{ DO } n$)

AKO ($A [p] == KOPUTA [(p+d)/2:n]$)

REZ [p] = 1;

ORREDUCE (REZ []);

CIRKULADNO

Kolika je složnost?

2. (MPI)

PROCES \emptyset

RECV ($a, *$);
 SEND ($a, 1$);
 $c = a$
 RECV ($b, *$);
 SEND ($b, 2$);
 $c = c - b$;

PROCESI 1, 2,

$a = ID + 1$;
 SEND (a, \emptyset);
 $c = a$;
 RECV ($b, *$);
 $c = c + b$;

1. VARIJANTA:

P#0

$a = 2$
 SEND ($a, 1$)

$c = 2$

SEND ($b, 2$)

$c = -1$

2. VARIJANTA

P#1

$a = 2$
 SEND ($a, 0$)

$c = 2$

$c = 4$

P#2

$a = 3$

SEND (a, \emptyset)

$c = 3$

$\rightarrow b = 3$

$c = 6$

$a = 3$

SEND (a, \emptyset)

$c = 3$

$\rightarrow b = 2$

$c = 5$

$a = 3$
 SEND ($a, 1$)

$c = 3$

$b = 2$

SEND ($b, 2$)

$c = 1$

$a = 2$

SEND (a, \emptyset)

$c = 2$

$b = 3$

$c = 5$