Parallelism nest week lab take computer produce () ? free.p(); add; ful. qr(); cors remove (); free. u() How are exam graded - is every semaphone's p &v called? - every senaphone has a value? - close it solve the problem? exercice & example Rapon(i from 1 to N) { How to parallelize this code? Déricle the computation into small sub-computations

Po = b. v [i]

P. = c. v [i]

problems: morth is embalanced between the Pz=g.V[i] · memory access to V(i) can be a problem as well Me could déruide our monk différently: for i from 4 to N &

// VV[i] = a + b. V[i] + c. V[i] & A.A.

// Do in parallel? + V[i] + c. V[i] + f. V[i] + g. V[i]

Do in parallel? + c. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i] + c. V[i]

Sc = a + b. V[i]

Sc = a = d+e.v[i]+f.v[i] +g.v[i]³) parallel

[300 end; i) = x + y x z

[> control parallelism: relies on the instructions to see what can be 11.

	Algorithmic complexity of this loop: O(N) for the	fust
	O(N)	
	on 3 prod	essors
	3) elle lake N mocessous	
	Po = VV[1] Complexity: O(1) Pr - VV[2] on N processors	
	Pr - VV[2] on N processors	
	PM - UNEWI	
	TNI	
,	from i from 1 to N in parallel	
	VVIII =	
	Chis is called Data parallelism.	
	you do not look at the profesam but at the array of	clata.
	3) Hormer sequences	
	VV[i]=a ×V[i](0+b/F°c[i]+di[i]3.	
	= a x V[i](o+ V[i)(b+ cv[i]+d·V[i] ² .	
	O(N) on 7 processors	
	This is a pipeline	
	Glarvii) - f.v[i]v.f wiii	
	20 mil had been as a feel a last to the	NI /
	If me don't have N processors but only of K	in proces.
	VV [] Each Ussoo My	nanta mi
	each your free for con	the last
	N/m colls	gun
	En data parallelism, hour do une associate some da	ta ta
	a processor? ©	10-10
	P. P	
	the park of the solle	
	bloc distribution cyclic distribution	
	€ could be unbalanced € you are messing	
	with the cache	
	보면 얼마나 나는 아니는 아니는 아니는 아니는 아니는 아니는 아니는 아니는 아니는 것이다.	

3

BARRED Wore than of parts bloc cyclic

you can play on the size of blocs to optimize for the cache untile reducing the probability of howing unbalanced computations.

mote: look at map reduce algorithm

PRAM

Parallel Random Acress Machine

. It has a set of processors, Pr. Pr., and each processor knows its id.

Each processor has a private memory and some registers. Program counter is THE register we can't do without ? - address of the next instruction to process)

* All processess have access to an unlimited shared memory.

All processors have "classical instructions +

- halt stops the processor

- fork : starts a new program on a free processor

· Each instruction in the P-RAM takes I und of time

· The execution model is synchronous: all processors performs

I instruction at the same time (1 step in P, takes 1 step in Ps): lockstep

· Read-Write conflicts

it depends (concurrent leads or write)

Read Wrik Concernent BCCR WE CW Exclusive REER WE EW

ER: No more than I access to a memory address

CR: Multiple accesses are allowed at the same time

EW: Exclusive white

CW: Allowed when all processors write the same realise other conditions: (1) consistency)

2) arbitrary only one realise will be written, but me don't know which one

3) priority: a processor has priority over others.

Size of B: (b)a) 2) given a node j: b has a size of 2m-1 - childs are 2j; 2j+1 - part is [i].	value	sion: applying a function to all evalues and the unite is the result. (Max, +,)
CREW 4 subtypes ERCW 4 subtypes CRCW 4 subtypes Exercise : How be nost cestwictive. Exercise : How Value on PRAM inpot : array of size n = 2 m output : Max = a[1]; for im a E if a [i] > max = a[i]; so im a E if a [i] > max = a[i	4 me	cintupes of P-RAM:
CREW 4 subtypes ERCW 4 subtypes CRCW 4 subtypes Exercise : How be nost cestwictive. Exercise : How Value on PRAM inpot : array of size n = 2 m output : Max = a[1]; for im a E if a [i] > max = a[i]; so im a E if a [i] > max = a[i		EREW for the rost of this course,
ERCW 4 subtypes CRCW A parallel algorithm is defined as a couple (algorithm, P-RAM), P-RAM being the most cestrictive. Exercise: Hax Value on P-RAM input: array of size n=2 ^m output: max value 1) Sequential vertica: V1) max=a[1]; for in a \(\frac{1}{2}\) if \		CREW we will study also on these 4P.RAM.
A parallel algorithm is defined as a couple (algorithm, P-RAM), P-RAM being the most cestrictive. Exercise: Hax Value on P-RAM input: array of size n=2 m output: max calue 1) Sequential vertion: V1) max=a[1]; for in a \(\)		ERCW 9 subtypes
Exercico: Max Value on P-RAM inpot: array of size n=2 ^m output: max value 1) Sequential Jertion: V1) max=a[1]; for in a \(\xi\) if a [i]> max \(\xi\) max=a[i]; is a [i] = max \(\xi\) vertur max; V2) for \(\xi\) from m-1 to \(\xi\) for \(\xi\) = \(\xi\) do \(\xi\) \(\xi\) = \(\xi\) \(\xi\) \(\xi\) \(\xi\) step 1 \(\xi\) \(\xi\) \(\xi\) step 1 \(\xi\) \(\xi\) step 2 \(\xi\) \(\xi\) step 3 \(\xi\) step 4 \(\xi\) \(\xi\) step 4 \(\xi\) \(\xi\) step 4 \(\xi\) \(\xi\) step 4 \(\xi\) step 5 \(\xi\) step 4 \(\xi\) step 5 \(\xi\) step 4 \(\xi\) step 6 \(\xi\) step 7 \(\xi\) step 1 \(\xi\) step 1 \(\xi\) step 2 \(\xi\) step 3 \(\xi\) step 4 \(\xi\) step 4 \(\xi\) step 5 \(\xi\) step 4 \(\xi\) step 6 \(\xi\) step 7 \(\xi\) step 8 \(\xi\) step 8 \(\xi\) step 9 \(\xi\) step 9 \(\xi\) step 1 \(\xi\) s		CRCWI
Exercico: Max Value on P-RAM inpot: array of size n=2 ^m output: max value 1) Sequential Jertion: V1) max=a[1]; for in a \(\xi\) if a [i]> max \(\xi\) max=a[i]; is a [i] = max \(\xi\) vertur max; V2) for \(\xi\) from m-1 to \(\xi\) for \(\xi\) = \(\xi\) do \(\xi\) \(\xi\) = \(\xi\) \(\xi\) \(\xi\) \(\xi\) step 1 \(\xi\) \(\xi\) \(\xi\) step 1 \(\xi\) \(\xi\) step 2 \(\xi\) \(\xi\) step 3 \(\xi\) step 4 \(\xi\) \(\xi\) step 4 \(\xi\) \(\xi\) step 4 \(\xi\) \(\xi\) step 4 \(\xi\) step 5 \(\xi\) step 4 \(\xi\) step 5 \(\xi\) step 4 \(\xi\) step 6 \(\xi\) step 7 \(\xi\) step 1 \(\xi\) step 1 \(\xi\) step 2 \(\xi\) step 3 \(\xi\) step 4 \(\xi\) step 4 \(\xi\) step 5 \(\xi\) step 4 \(\xi\) step 6 \(\xi\) step 7 \(\xi\) step 8 \(\xi\) step 8 \(\xi\) step 9 \(\xi\) step 9 \(\xi\) step 1 \(\xi\) s	A	narallel algorithm is defined as a counte
inpot: array of size n=2 ^m output: max value 1) Sequential vertion: V1) max=a[1]; for im a \(\xi \) if a [i] max \(\xi \) max=a[i]; setun max; V2) for \(\xi \) from m-1 to 0 for \(\xi = 2^{\xi} \) to \(2^{\xi \cdot 1} \), \(\xi \) [2] +1]); sep1 step2 step2 step3 step4 of from the nodes from the soft, breadth first Size of \(\xi \): (\xi \) a) 2) given a node \(\xi \): h has a size of \(2 \xi \) - childs are \(2 \xi ; 2 \xi \) - yearnt is \(\xi \) 2].	(algorith	m, P-RAM), P-RAM being the most restrictive.
1) Sequential Jersion: V1 max=a[1]; for in a \(\) if a \(\) if a \(\) in ax \(\) max = a[i]; \[\frac{3}{3} \] \[\text{detun max;} \] V2) for \(\) from \(\) m-1 to \(\) \[\text{for } \)	Exercico:	Hax Value on P-RAM
1) Sequential Jersion: V1 max=a[1]; for in a \(\) if a \(\) if a \(\) in ax \(\) max = a[i]; \[\frac{3}{3} \] \[\text{detun max;} \] V2) for \(\) from \(\) m-1 to \(\) \[\text{for } \)	input	· array of size n=2m
1) Sequential Jersion: V1 max=a[1]; for in a \(\xi \)	outpu	t: max value
Joseph Johnson Joseph J		
for in a \(\) if a \(\) is a \	1) Segu	ential Jertion:
if a [i]) max \{ max = a [i] 3 3 Letun max; V2) for \(\) from m-1 to \(0 \) \(\) for \(\) from m-1 to \(0 \) \(\) for \(\) from m-1 to \(0 \) \(\) for \(\) from m-1 to \(0 \) \(\) for \(\) from m-1 to \(0 \) \(\) for \(\) from m-1 to \(0 \) \(\) for \(\) from the upulsent a \(\) dep \(\) \(\) step 1 \(\) from the nodes from the soft, breadth first \(\) Size \(\) of \(\) : \(\) (\(\) \(\) a \) \(\) dep \(\) from the node \(\) : \(\) has a size of \(2 \) m-1 \(\) - fraunt is \(\) \(\) 2 ; 2 ; 1 1 \(\) - fraunt is \(\) \(\) 2.	V1	max: al1];
max = a[i] 3 Leturn max; V2) for l from m-1 to 0 for j = 2 do 2 etd-1 in parallel Lij] = max (l(2j]; b[2j+1]); step 1 step 2 step 2 tree as an array? step 3 step 4 from the nodes from the step 4 from the noot, breadth first Size of l: (l)a) l has a size of 2 m-1 - fraunt is [i] have a size of 2 m-1 - fraunt is [i]		
Jetun max; V2) for l from m-1 to 0 for j=2 do 2 l+1-1 in parallel lij] - max (l(2j]; b[2j+1]); step1 tree as an array? step2 tree as an array? step3 step3 step4 of line the nodes from the local from the node from the node, breadth first Size of l: (l)a) 2) given a node j: l has a size of 2 m-1 - childs are 2j; 2j+1 - parent is [i].		
for l from m-1 to 0 for j=2 do 2 l+1-1 in parallel li [j] = max (l(2j]; b[2j+1]); step1 step2 step3 step3 step4 of from the nodes from theret from the node, breadth first Size of l: (l)a) l has a size of 2 m-1 - childs are 2j; 2j+1 - parent is [i].		max=alil
for l from m-1 to 0 for j=2 do 2 l+1-1 in parallel li [j] = max (l(2j]; b[2j+1]); step1 step2 step3 step3 step4 of from the nodes from theret from the node, breadth first Size of l: (l)a) l has a size of 2 m-1 - childs are 2j; 2j+1 - parent is [i].		3
step? step? step? step? step? step.4 1) number the nodes from the first Size of b: (b)a) b has a size of 2 m - 1 - paent is [i].	1	setur max;
step? step? step? step? step? step.4 1) number the nodes from the first Size of b: (b)a) b has a size of 2 m - 1 - paent is [i].	1121	
step? step? step? step? step? step.4 1) number the nodes from the first Size of b: (b)a) b has a size of 2 m - 1 - paent is [i].	VZ	for them m-1 to 0
step? step? step? step? step? step.4 1) number the nodes from the first Size of b: (b)a) b has a size of 2 m - 1 - paent is [i].		1 5: 7 mar (DE2: 7 h [2: 417);
step? step? step? step? 1) number the nodes from the sort, breadth first Size of B: (b)a) 2) given a node j: 1 has a size of 2 m - 1 - paunt is [i].		et [] = maa (a(2] 1; 0(2] (3))
step? step? step? step? 1) number the nodes from the sort, breadth first Size of B: (b)a) 2) given a node j: 1 has a size of 2 m - 1 - paunt is [i].	ster 1	How to upresent a
Size of B: (Da) 2) given a node j: I has a size of 210-1 - paent is [17.		tree as an array?
Size of B: (b)a) 2) given a node j: b has a size of 2m-1 - childs are 2j; 2j+1 - part is [i].	steps step4	1) number the nodes from the
- paint is 3 .	901	from the nort, breadth first
- paint is 3 .	Size	of b: (b)a) 2) given a node j:
- paint is 3 .	& has	a size of 210-1 - childs are 2; 2; 11
	0 5	- paint is 127.
Nax n 2n-1	6	n $2n-1$

1	
	On which P-RAM does it work?
	→ No processor writes in the same cell
	→ No processor reads in the same cell
	L) so seve need an EREW P-RAM.
	Complexity: O(log(Nu)) on n processors
	Complexity: O(log(Nu)) on n processors 40(n) (ling Onotation
	For mext une la
	1) Is conclusite of the securation of the aboutton
	2) you are free to use the P-RAM you want, what is the fastest algorithm to compute the max value?
	aldouthou to complete the more states?
	argentin so conquire the suax valle?
	In Ord space
	Tou lab session
	OPENMP -> lilerary (ext-of the compiler) "sindicate where shit's parallel and it deals with
	thread Comment ships present and it oreas such
	threads for you.
_	An open MP program is single process with multiple threads
	(simulates a P-RAM)
J- 1	Threads are created when encontering a parallel region.
	The number of threads is decided at one time runtime by
_	OpenMP, luit can be set by developpers.
	Each thread has an id and the mader thread has an id of o.
_	the gradina only paravel too
	parallel foi(i=0;i<10000;i++) ?
	Suschrenia 3
	HOW TO COMPAL.
	gcc -o test-para test.c -fopenmp
_	parallel 2
	install acc: Linux Subsystem (installed last years)
_	(installed last years)