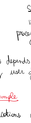
25 Lezione

131.121.132.61

Subject: Open CL access

1° Example

You have a non optimal assembly machine.
You have a parallel degree: m_{max}



(6:40)

If 1st application
the time is well
work time than before.

Supposing that some other time is free I can unlock this process from this core and assign it to another core.

this depends on the fact that you are not the only user of the machine.

2° Example

Applications with "phases".
↳ tempo delle conversione inizio in diversi momenti della giornata.

Can you fix yourself a choice?

a) Variare il parallelism degree
(Continuing modelling)

from
map

Moving from a situation where I have 3 workers to a situation where I have 5 workers.

If I have a farm

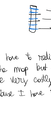
$$farm(f, n_w) \quad n_w > n_{max}$$
$$\downarrow$$
$$farm(f, n_{max})$$

I have to find the another and say to the another that the workers are more than before.

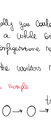
If I have a map and I want to move to a map with more workers:

$$map(f, n_w) \quad n_w > n_{max}$$
$$\downarrow$$
$$map(f, n_{max})$$

I have to split the data in a different way.



then I decide to add 2 workers



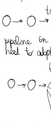
I have to redistribute.
the map but this can be very costly and also isn't easy because I have to stop the threads.

(6:50)

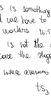
Totally you could have some code that for a whole computer cores, or for reconfiguration run a free slide. (6:54)

All the workers must agree to redistribute.

Pattern example



the problem in become slowness and we had to adapt.



this is something that requires some time and we have to interrupt the sending the workers (6:59)
this is not the same tasks as before because the stage 1 is not affected. (6:59)

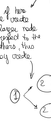
So we assume that

$$t_{s_1} = k$$
$$\downarrow$$
$$\frac{k}{4}$$

t_{s_1} become much faster than before
Now where is the bottleneck?

↳ I create a single stage with the combination of the stage 2 and 3 of the stage 2

if now we create a large node in respect to the others, this may create



Now for to model the graph 5 through in Computer Science we ignore 3.

$C_1 \longrightarrow C_3$

C_1	C_2	C_3
t_1	t_2	t_3
t_3	t_4	

The performance of this two threads that share a core are not the performance of a single thread ?? (6:13)

Socket 1				Socket 0			
C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
t_1	t_2	t_3	t_4	t_5			
	f	f		f			

Continuously the frequency of one core Combine under the frequency for both a thread on each core (on turbo)

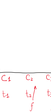
Quality the process for it work be BFS.

Provide different an pattern, divide under the max parallel degree.

there is the possibility to declare a farm and use the farm with a max number of worker that is 10.

the actual max is 100.

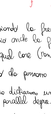
You have a system:



(7:17)

Look at the sensors and if there is something wrong talk to actuators to change the behaviour.

How to program this kind of things?



MAPE: allow for practice actions

If I have:

STABILITY PROBLEM

changes are small but frequent
↳ never reach the steady state

farm(10) to workers are too much

$$\downarrow$$

$$t_A > \frac{t_{MW}}{10}$$

$$farm(3) \quad t_A < \frac{t_{MW}}{3}$$

Whatever the tradeoff I choose I can go for 0.
(7:30)

In general 5 a good idea to ensure stability that the decisions are...

If I increase the number of workers is a bad idea to decrease drastic the change.