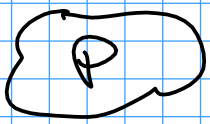
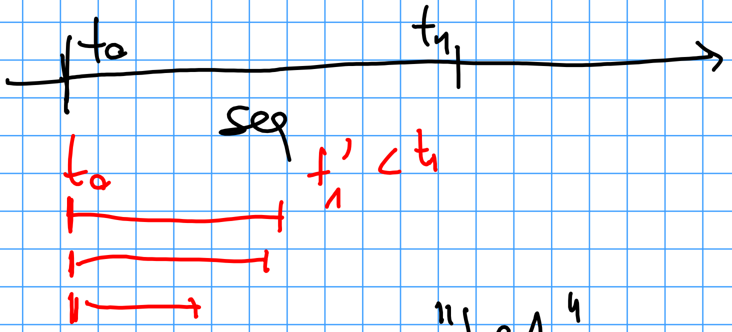
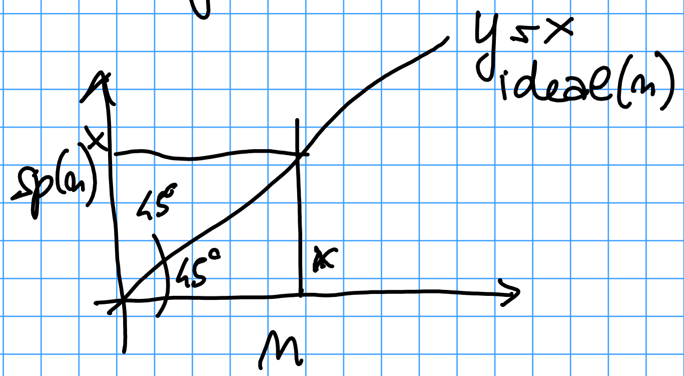




"best" you know

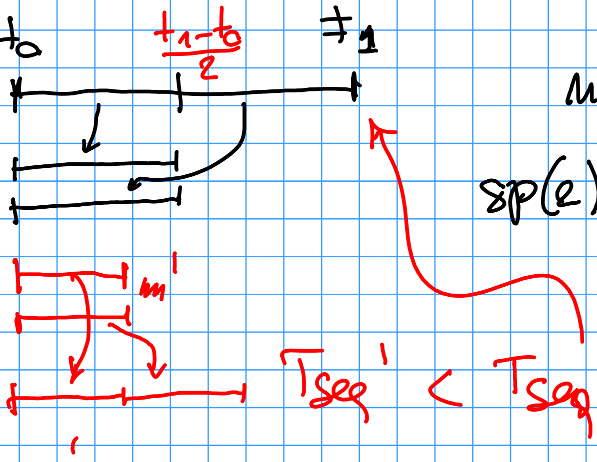
$$speedup(n) = \frac{T_{seq}}{T_{par}(n)}$$

↑
par degree



why idealsp(n) = n ?

prob (seq) → uses k Mbytes in the working set



n=2

sp(2) = 2

Tseq' < Tseq

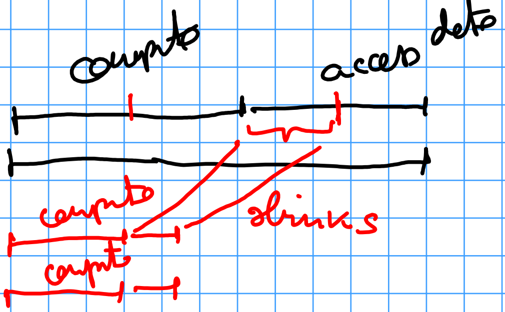
sp(2) = 2.xx > 2

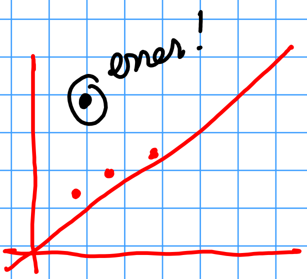
L1 cache of my core $\frac{k}{2}$ Mbytes

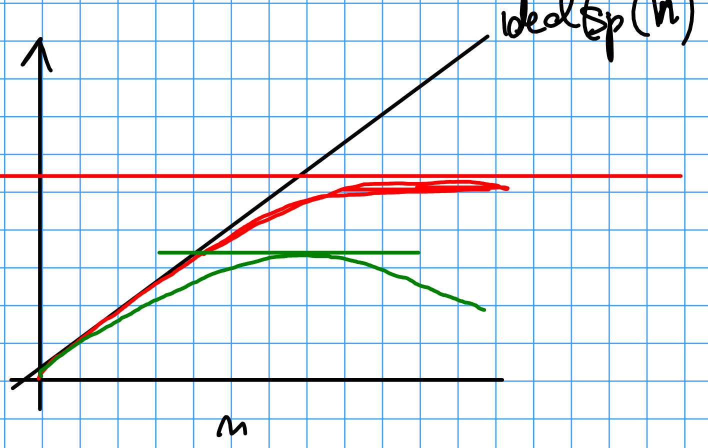
Tseq' iff each "half" of comp uses half of the WS

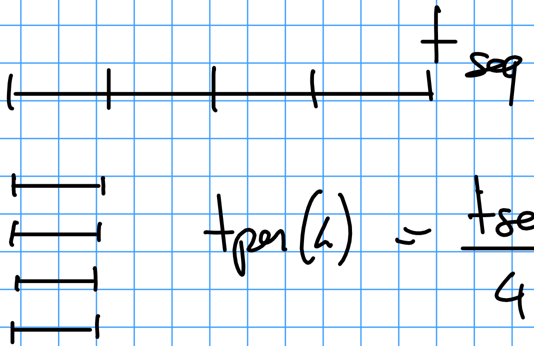
⟹ WS fits L1 cache

seq

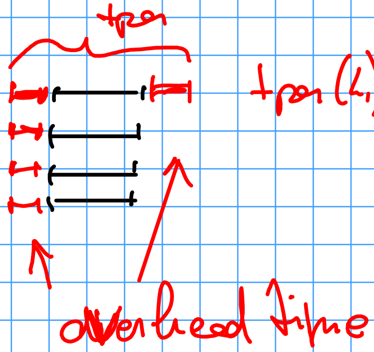






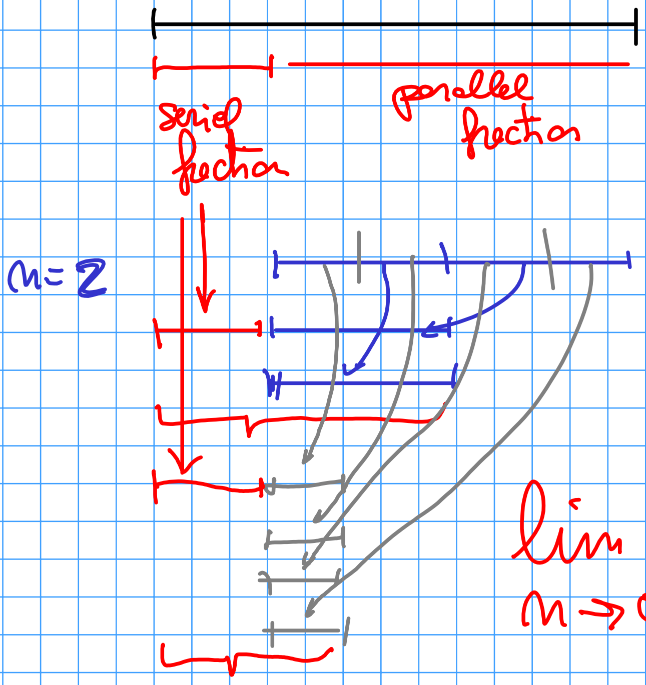


$$t_{par}(4) = \frac{t_{seq}}{4}$$



$$t_{par}(4) > \frac{t_{seq}}{4}$$

Amdahl Law



$(T_{ser} + T_{non\,ser})$

$=$

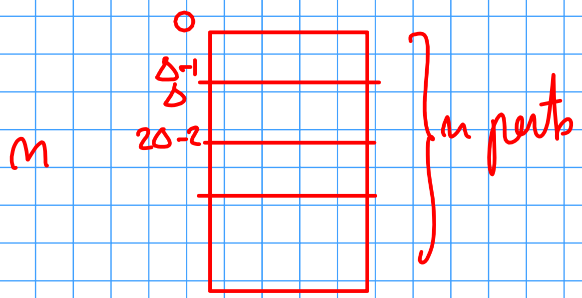
$$\underset{n\to\infty}{Sp(n)} = \frac{T_{seq}}{\boxed{T_{ser}}}$$

$$\lim_{n\to\infty} T_{par}(n) = T_{serial\ fraction}$$

T_{seq} $f \in [0,1]$ serial fraction $(1-f)$ non serial fraction

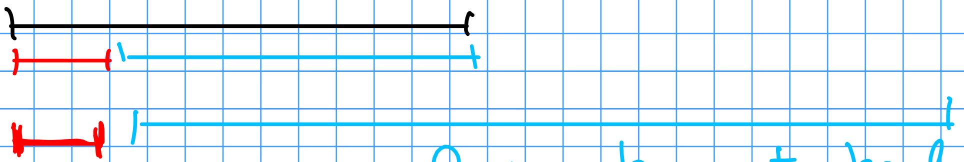
$$Sp(n) = \frac{T_{seq}}{f\,T_{seq} + \frac{(1-f)\,T_{seq}}{n}} \overset{n\to\infty}{=} \frac{\cancel{T_{seq}}}{f\,\cancel{T_{seq}}} = \frac{1}{f}$$

serial fraction doesn't increase with the size of the data

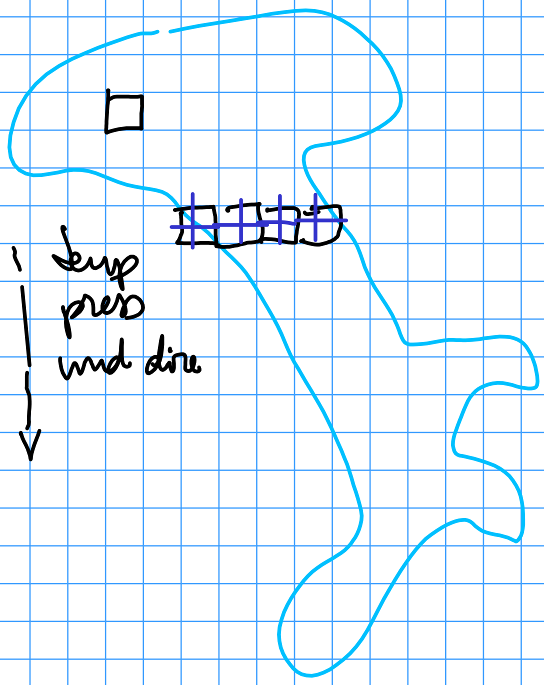


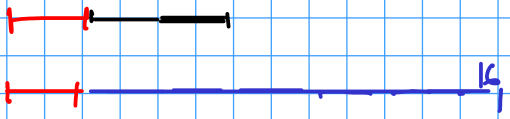
$$\rightarrow \text{thread } i \begin{cases} i \cdot \Delta \\ ((i+1)\cdot\Delta - 1) \end{cases}$$

$$\frac{len}{n} = \Delta$$



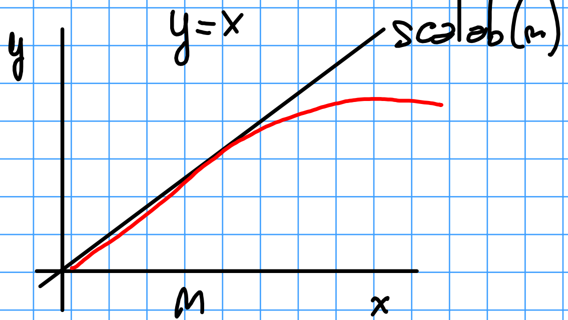
larger work to handle





scalability

$$\underline{\underline{\text{scalab}}}(n) = \frac{T(1)}{T(n)}$$



Efficiency

$$\varepsilon(n) = \frac{T_{id}}{T(n)}$$

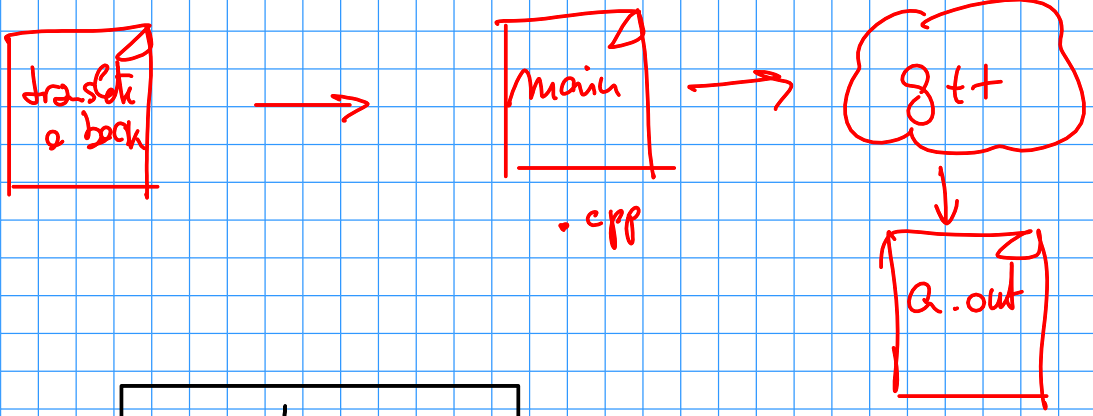
← ideal parallel time

← actual parallel time

$$T_{id} = \frac{T_{seq}}{n}$$

$$\varepsilon(n) = \frac{T_{seq}}{n\,T(n)} = \frac{\text{speedup}(n)}{n}$$

std :: chrono



```
> time a.out ....
```

std :: chrono

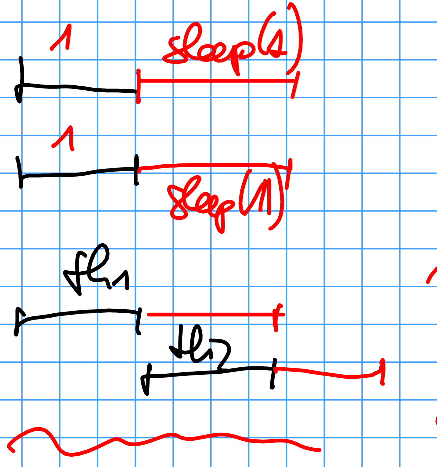
```
t0 = now();
≡
t1 = now();
elapsed = fn( t1 - t0 )
```

$texe_1$ . . . $texe_n$ ⇒ avg

12 13.1 12.5 12.7 ⑰ ⑩

```
> a.out ....  1024
n 10 s 12.7 ..... 1024 XXX
```

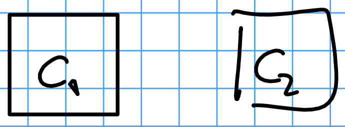
n	1	S	1.2	1024	xxx
n	1	S	1.3	"	"
n	1	S.	1.4	-	-



2 sec

2sec

3 sec rather than 4



stick (th1 , c1)
(th2 , c2)

```
X { x = 1.234
    for(        )
         X = sin(x);   }

cout << x ;
```

-O3