Loi d’Amdahl

https://fr.wikipedia.org/wiki/Loi_d%27Amdahl

L’optimisation est limitée : on ne peut pas aller plus de 20 fois plus vite même avec 95% de parallélisme

$$\cancel{SpeedUp = \frac{1}{(1-s)+\frac{s}{Ac}}}$$

T = temps d’exec. du programme avant tentative d’amélioration

T_a = temps d’exec. après amélioration par parallélisation

s=(T-Ta)/Ta

Ac = Accélération

Loi d’Amdahl

Présentation issue du document suivant : http://www.cristal.univ-lille.fr/~marquet/cnl/ppd/intro.pdf (voir page 31 et 32)

Un programme donné peut se décomposer en une partie purement séquentielle (non parallélisable) et une partie parallélisable, soit :

T = alpha*T + (1-alpha)*T

où alpha*T est le temps de travail purement séquentiel (coût fixe quel que soit le nombre d’UC)
et (1-alpha)*T le temps de travail de la partie parallélisable.

Si on améliore l’infrastructure pour exécuter le programme sur p UC, le temps d’exécution sera donc :

Tp = alpha*T + (1-alpha)/p*T

Le SpeedUp sera donc :

SpeedUp = T/Tp = $\frac{T}{alpha*T+T*\frac{(1-alpha)}{p}}=\frac{1}{alpha+\frac{(1-alpha)}{p}}=\frac{p}{alpha*p+1-alpha}$

quand p ->∞ alors SpeeUp -> $\frac{1}{alpha}$

On retrouve un comportement conforme au schéma wikipédia :
https://fr.wikipedia.org/wiki/Loi_d%27Amdahl#/media/File:Ley_de_Amdahl.svg

En effet :

Taux de code parallélisable (1-alpha)	SpeedUp maximum possible (si p->infini alors SpeeUp=1/alpha)
50%	1 / 0.5 = 2
75%	1 / 0.25 = 4
90%	1 / 0.1 = 10
95%	1 / 0.05 = 20

Remarque : dans le calcul du SpeedUp max., on ne prend pas en compte les latences supplémentaires relatives aux temps de préparation, transfert et réception des données échangées entre les UC pour produire le résultat final et qui pénalisent encore SpeedUp.

Loi d'Amdahl - un exemple de calcul

+ - * /   1 UT # On suppose que les op. élémentaires prennent 1 unité de tps

=   1 UT   # On suppose affectation = 1 UT

F 1 "+" et 1 "=" 2 UT

G 1 "+" et 1 "*" et 1 "=" 3 UT

Prendre en compte le temps passer à transmettre des résultats intermédiaires entre unités de calcul

PE1 UT   # Temps de préparation des émissions de données

T/P1 UT # Temps de propagation/transmission

R1 UT # Temps de réception

A: x=1

B: y=2

C: z=F(x)

D:w=G(z)

E: result=z+w+y

Sur 1 processeur :

	A	B	C	C	C	D	D	D	D	E	E	E	
0	1	2	3	4	5	6	7	8	9	10	11	12	13

On se demande combien de temps cela prendrait avec deux processeurs. Il faut connaître le degré de parallélisation.

Sur 2 processeurs :

Peu optimisé

m2	B						R	D	D	D	D	PE						
						T							T					
m1	A	C	C	C	PE									R	E	E	E	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

Un peu mieux optimisé

m2	B	PE																
			T															
m1	A	C	C	C	R	D	D	D	D	E	E	E						
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

Sur cet exemple, on ne parvient pas à faire mieux que l’exécution sur 1 processeur. :-(

Nouvel exemple:

A: x=1
B: y=2
C: z=F(x)
D: w=G(y)
E: resultat = z+w
→ sur une seule machine: 11 UT (Unités de temps)

Sur 1 processeur :

m1	A	B	C	C	C	D	D	D	D	E	E							

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

Sur 2 processeurs :

m2	B	D	D	D	D		R	E	E	
						T				
m1	A	C	C	C	PE					
0	1	2	3	4	5	6	7	8	9	10

Calcul du speedup :

$$T = 11$$
$$T_a = 9$$
$$SpeedUp = \frac{T}{Ta} = \frac{11}{9} = 1.22$$

$$T_a = (1-s)T + \frac{sT}{A_\epsilon}$$

avec s=2/9 = fraction du temps perdu a faire de l'optimisation (les opérations PE ou R ou T mais sans A/B/C/D/E en parallèle)

$$A_\epsilon = \frac{sT}{T_a - (1-s)T} = \frac{s}{1/s + s - 1} = \frac{s^2}{s^2 - s + 1} = 1.17$$
$$A_\epsilon = \frac{sT}{T_a - (1-s)T} = \frac{s}{SpeedUp^{-1} + s - 1} = 5.545$$

D'autres lois existent, et prennent en compte d'autres variables, par exemple la consommation électrique des machines.

$$SpeedUp = \frac{1}{(1-s) + \frac{s}{Ac}}$$

T = temps d'exec. du programme sur une seule UC
T_a = temps d'exec. après parallélisation

$$\cancel{s = \frac{(T - Ta)}{Ta}}$$

Ac ???

Loi d’Amdahl - un exemple de calcul

$+\ -*/\ \leftrightarrow\ 1\,UT$ # On suppose que les op. élémentaires prennent 1 unité de tps
$=\ \leftrightarrow\ 1\,UT$ # On suppose affectation = 1 UT

$$\cancel{F \leftrightarrow 1\,"+"\,et\,1\,"="\ \leftrightarrow\ 2\,UT}$$
$$\cancel{G \leftrightarrow 1\,"+"\,et\,1\,"*"\,et\ 1\,"="\ \leftrightarrow\ 3\,UT}$$

Prendre en compte le temps passer à transmettre des résultats intermédiaires entre unités de calcul
$PE \leftrightarrow 1\,UT$ # Temps de préparation des émissions de données
$T/P \leftrightarrow 1\,UT$ # Temps de propagation/transmission
$R \leftrightarrow 1\,UT$ # Temps de réception

$$\cancel{A{:}\ x = 1}$$
$$\cancel{B{:}\ y = 2}$$
$$\cancel{C{:}\ z = F(x)}$$
$$\cancel{D{:}\ w = G(z)}$$
$$\cancel{E{:}\ result = z + w + y}$$

~~Sur 1 processeur :~~

	~~A~~	~~B~~	~~C~~	~~C~~	~~C~~	~~D~~	~~D~~	~~D~~	~~D~~	~~E~~	~~E~~	~~E~~	
~~0~~	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~	~~6~~	~~7~~	~~8~~	~~9~~	~~10~~	~~11~~	~~12~~	~~13~~

~~On se demande combien de temps cela prendrait avec deux processeurs. Il faut connaître le degré de parallélisation.~~

~~Sur 2 processeurs :~~

~~Peu optimisé~~

~~m2~~	~~B~~						~~R~~	~~D~~	~~D~~	~~D~~	~~D~~	~~PE~~						
						~~T~~							~~T~~					
~~m1~~	~~A~~	~~C~~	~~C~~	~~C~~	~~PE~~									~~R~~	~~E~~	~~E~~	~~E~~	

~~0~~	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~	~~6~~	~~7~~	~~8~~	~~9~~	~~10~~	~~11~~	~~12~~	~~13~~	~~14~~	~~15~~	~~16~~	~~17~~	~~18~~

~~Un peu mieux optimisé~~

~~m2~~	~~B~~	~~PE~~																
			~~T~~															
~~m1~~	~~A~~	~~C~~	~~C~~	~~C~~	~~R~~	~~D~~	~~D~~	~~D~~	~~D~~	~~E~~	~~E~~	~~E~~						
~~0~~	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~	~~6~~	~~7~~	~~8~~	~~9~~	~~10~~	~~11~~	~~12~~	~~13~~	~~14~~	~~15~~	~~16~~	~~17~~	~~18~~

~~Sur cet exemple, on ne parvient pas à faire mieux que l'exécution sur 1 processeur. :-(~~

~~Nouvel exemple:~~

~~A: x=1~~
~~B: y=2~~
~~C: z=F(x)~~
~~D: w=G(y)~~
~~E: resultat = z+w~~
~~→ sur une seule machine: 11 UT (Unités de temps)~~

~~Sur 1 processeur :~~

~~m1~~	~~A~~	~~B~~	~~C~~	~~C~~	~~C~~	~~D~~	~~D~~	~~D~~	~~D~~	~~E~~	~~E~~							
~~0~~	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~	~~6~~	~~7~~	~~8~~	~~9~~	~~10~~	~~11~~	~~12~~	~~13~~	~~14~~	~~15~~	~~16~~	~~17~~	~~18~~

~~Sur 2 processeurs :~~

~~m2~~	~~B~~	~~D~~	~~D~~	~~D~~	~~D~~		~~R~~	~~E~~	~~E~~	
						~~T~~				
~~m1~~	~~A~~	~~C~~	~~C~~	~~C~~	~~PE~~					
~~0~~	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~	~~6~~	~~7~~	~~8~~	~~9~~	~~10~~

~~Calcul du speedup :~~

$$T = 11$$
$$T_a = 9$$
$$SpeedUp = \frac{T}{Ta} = \frac{11}{9} = 1.22$$

$$T_a = (1-s)T + \frac{sT}{A_{\epsilon}}$$

~~avec s=2/9 = fraction du temps perdu a faire de l'optimisation (les opérations PE ou R ou T mais sans A/B/C/D/E en parallèle)~~

$$A_{\epsilon} = \frac{sT}{T_a - (1-s)T} = \frac{s}{1/SpeedUp + s - 1} = \frac{s^{2-}}{s^2 - s + 1} = 1.17$$
$$A_{\epsilon} = \frac{sT}{T_a - (1-s)T} = \frac{s}{1/SpeedUp + s - 1} = 5.545$$

~~D'autres lois existent, et prennent en compte d'autres variables, par exemple la consommation électrique des machines.~~