# Bande passante

- compiler `pratique` et `pratique.FM`
- profiler (`nvprof`) les noyaux et conclure
- (Tesla T4) : enlever `exp` dans les noyaux
- mesurer les temps et calculer les bandes passantes effectives grâce à

$$\frac{\#R_{ker} + \#W_{ker}}{10^9 t}$$

- le nombre d'écriture et de lecture est en octets !
  - comparer avec la bande passante theorique (`theo.x`)
  - décommenter `--ptxas-options=-v` dans le `Makefile`...
  - .. et recompiler `make 2>&1 | c++filt` →que dit le compilo ?

# Memoire verrouillée

- compiler `pinned.cu`
- profiler : `nvprof --print-gpu-trace ./pinned.x`
- regarder les colonnes `SrcMemType, TroughPut, Duration`
- qu'en déduisez vous ?
- pour les courageux
  - changer `cudaHostAllocDefault` par `cudaHostAllocMapped`
  - supprimer les transferts
  - ecrire un noyau simple, e.g. homothetie
  - verifier la modif de la mémoire sur l'hôte

# Transferts asynchrones

- compiler et executer exAsync
- verifier que la version asynchrone est plus courte
- profiler : `nvprof --print-gpu-trace ./exAsync.x`
- reperez les differents appels
- essayer de changer le nombre de morceaux
- génerer une trace

  `nvprof --output-profile async.prof ./exAsync.x ...`
- ... et la rapatrier pour visualiser avec `nvvp async.prof`
- (bonus) passer le noyau en 2D, découpé en morceaux selon $x$