Si ma mémoire est bonne...

Un tour d'horizon matériel et logiciel
des mémoires informatiques
présentes et futures.

Les mémoires actuelles : Le cache SRAM

- Le temps d'accès et le débit suivent la vitesse du processeur avec une granularité d’un mot mémoire.

- La taille de la SRAM est limitée par sa consommation électrique, car chaque bit nécessite cinq transistors qui doivent être alimentés continument.

- Cache Coalescence : essayer de regrouper le maximum de valeurs utiles quand on accède à une ligne de cache.

- Cache reuse : essayer d'accéder aux données déjà stockées dans les caches le plus tôt possible avant qu'elles ne soient évincées.

Les mémoires actuelles : La mémoire principale DRAM

- Le temps d'accès stagne depuis 30 ans autour de 50 à 100 ns.

- Pour compenser, on augmente la taille d'un mot mémoire : 128 octets sur une barrette de DDR4 (16 octets pour la DDR1 en l'an 2000).

- Ce mot passe par un bus reliant la mémoire et le CPU de plus faible largeur (64 bits), mais de plus haute fréquence (2,66 GHz) à ne pas confondre avec la largeur et fréquence d'accès de la mémoire elle-même.

- On atteint une limitation des puces multicœurs, car la complexité du composant est proportionnelle au nombre de cœurs x nombre de bus mémoire (AMD EPYC : 32 x 8).

Les mémoires actuelles : Le stockage sur disque dur

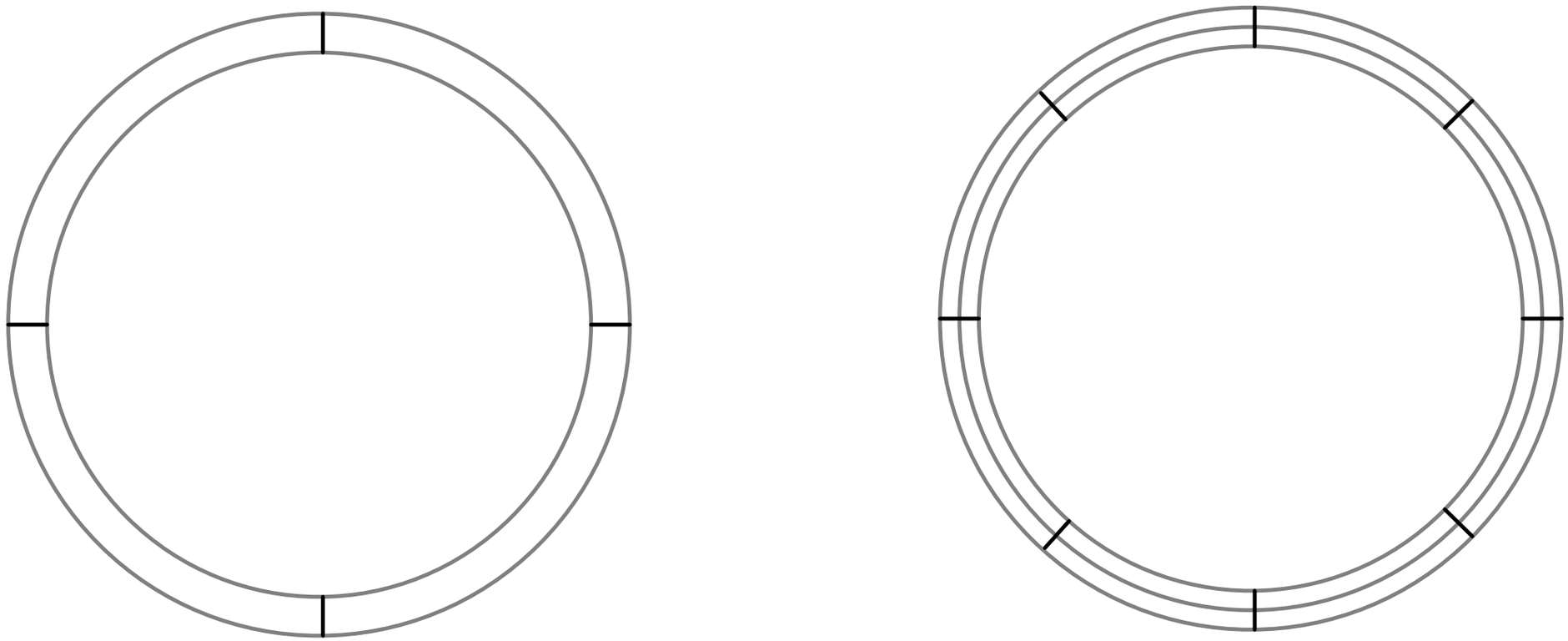
- Régulièrement l'épaisseur et la longueur du substrat chimique stockant un bit diminuent ce qui augmente d'autant le nombre de pistes par plateau et le nombre de secteurs sur une piste.

- La capacité de stockage d'un disque dur à plateaux est égale au nombre de secteurs/piste multiplié par le nombre de pistes.

- Mais la vitesse de lecture / écriture ne dépend que du nombre de secteurs lus en une rotation du plateau, dont la vitesse stagne pour des raisons mécaniques (5400 à 15000 Tpm), et par conséquent le débit croit avec la racine carrée de la taille !

- Le temps d'accès est lié à la vitesse rotationnelle et est lui aussi bloqué à 5 ou 10 ms.

Les mémoires actuelles :
Le stockage sur disque dur



- Secteurs par piste = x2
- Pistes par disque = x2
- Taille = x4
- Vitesse = x2

Mémoires futures : High Bandwidth Memory

- DRAM : fondue en même temps que le CPU ou GPU, elle très proche, donc rapide, mais de taille limitée à 64 GO à ce jour.

- Grand nombre de puces de DRAM empilées physiquement et connectées par micro mécanique et micro soudures.

- Latence un peu plus faible que la DRAM, car elle est située à quelques millimètres du processeur.

- Très gros débit grâce à la largeur de bus encore plus grande que la DDR4 externe : 4096 bits -> 1 TO/s.

- C'est aujourd'hui la seule voie pour poursuivre l'augmentation du nombre de cœurs de CPU ou unités de calcul des GPU.

Mémoires futures : Solid Stade Drives

- Le prix au TO baisse plus vite que celui des disques durs, on en trouve à 200€.

- Faible endurance d'écriture : ne peut pas servir de cache disque ou de mémoire swap.

- Latence de 10µs à 100µs et débit de 500 MO/s à 3 GO/s, intermédiaire entre le disque dur et la DRAM.

- Le goulot d'étranglement des I/O n'est plus le matériel, mais le logiciel : format ASCII, décodage XML, GLIBC séquentielle limitée à 500 MO/s, pré/post processing des données.

Mémoires futures : 3D XPoint

- Nouveau type de mémoire mis au point par Intel et Micron Technologie qui est un hybride de DRAM et de SSD.

- Prix et performance entre la DRAM et le SSD.

- Le plus gros avantage est sa grande longévité en écriture qui lui permet d'être utilisée comme buffer disque, mémoire SWAP ou même comme mémoire principale.

- Vitesse maximale obtenue avec une plus faible granularité que les SSD : 4 KO, idéal pour le SWAP.

- Utilisée comme un disque de SWAP rapide ou bien comme une barrette de mémoire vive plus lente.

Mémoires futures : Disques durs améliorés

- Shingle Magnetic Recording (PMR/SMR) : aujourd'hui x3, jusqu`à x10, mais nécessite de lire et réécrire un groupe de pistes à la fois ce qui rend l'écriture non aléatoire.

- Heat Assited Magnetic Recording : en phase préindustrielle, potentiel x10, le laser préchauffe la piste avant l'écriture ce qui ralentit son temps d'accès (100ms).

- Bit Patterning : idem encodage linéaire actuel, mais avec un pattern en 2D, donc nécessite de lire et réécrire un groupe de pistes simultanément.

- En 2030 : disques durs de 1 PO, mais très contraints en écriture donc idéal pour de l'archivage ou des accès en flux.

Contraintes logicielles : Lectures/écritures par bloc

- DRAM : bloc de 128 octets.
- HBM : bloc de 1 Ko.
- 3D Xpoint : bloc de 4 Ko.
- SSD : bloc de 128 Ko.
- HD actuels : optimum pour des blocs > 1 Mo.
- HD futurs : optimum pour des blocs > 100 Mo.

Contraintes logicielles : Réutilisation des caches

- Chaque niveau de mémoire a une granularité plus fine que celle du dessous

- Cela permet de stocker des informations en attente d'écriture afin de les regrouper pour utiliser la bande passante du niveau suivant de manière efficace.

- En lecture cela permet de retenir les données qui viennent d'être lues de manière consécutive, mais ne sont pas encore traitées.

Solutions logicielles : Changement de dimension

- Toutes les mémoires sont unidimensionnelles.
- Les structures de données sont multidimensionnelles.
  - Dimension fixe et topologie régulière : bitmaps, vecteurs, matrices, tableaux.
  - Dimension fixe, mais topologie irrégulière : maillages, arbres.
  - Dimension et topologie arbitraire : graphes.

Solutions logicielles : Renumérotation des données

- Les Space Filing Curves projettent un espace multidimensionnel sur un espace unidimensionnel.

- Elles permettent de linéariser au mieux les accès mémoire lorsqu'on parcourt une structure de donnée multidimensionnelle de manière aléatoire.

- Les courbes les plus complexes sont de nature géométrique et moins évidente à transposer sur des graphes arbitraires, mais idéale pour des données graphiques, des matrices ou des maillages.

- La courbe en Z est très simple à implémenter en dimension arbitraire : il s'agit d'entrelacer les bits des coordonnées dimensionnelles en un seul super entier.

- Exemple : modifier un bloc de 3x4 cases dans un tableau de 8x8 avec des lignes de cache de 8 valeurs consécutives.

Numérotation en LIGNE
4 défauts de cache
1-8, 9-16, 17-24 et 25-32

57	58	59	60	61	62	63	64
49	50	51	52	53	54	55	56
41	42	43	44	45	46	47	48
33	34	35	36	37	38	39	40
25	26	27	28	29	30	31	32
17	18	19	20	21	22	23	24
9	10	11	12	13	14	15	16
1	2	3	4	5	6	7	8

Numérotation en COLONNE
3 défauts de cache
41-48, 49-56 et 57-64

8	16	24	32	40	48	56	64
7	15	23	31	39	47	55	63
6	14	22	30	38	46	54	62
5	13	21	29	37	45	53	61
4	12	20	28	36	44	52	60
3	11	19	27	35	43	51	59
2	10	18	26	34	42	50	58
1	9	17	25	33	41	49	57

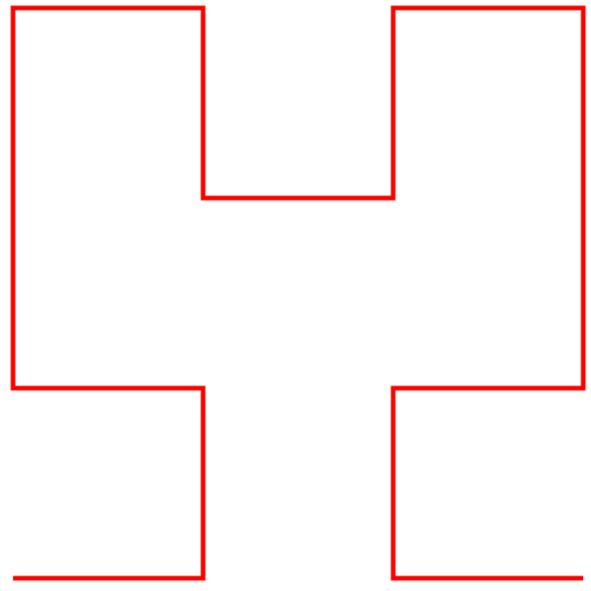
Numérotation FRONTALE
3 défauts de cache
25-32, 33-40 et 49-56

64	63	62	61	60	59	58	57
49	48	47	46	45	44	43	56
36	35	34	33	32	31	42	55
25	24	23	22	21	30	41	54
16	15	14	13	20	29	40	53
9	8	7	12	19	28	39	52
4	3	6	11	18	27	38	51
1	2	5	10	17	26	37	50

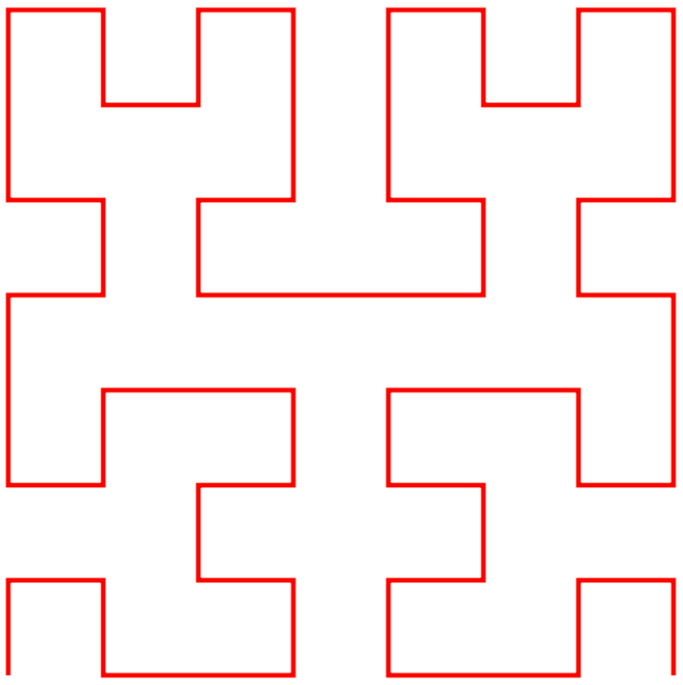
Numérotation de HILBERT
2 défauts de cache
49-55 et 59-64

22	23	26	27	38	39	42	43
21	24	25	28	37	40	41	44
20	19	30	29	36	35	46	45
17	18	31	32	33	34	47	48
16	13	12	11	54	53	52	49
15	14	9	10	55	56	51	50
2	3	8	7	58	57	62	63
1	4	5	6	59	60	61	64

Itération 1



Itération 2



Itération 3

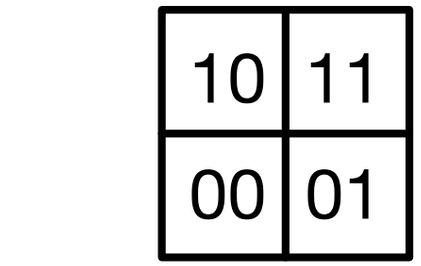
22	23	26	27	38	39	42	43
21	24	25	28	37	40	41	44
20	19	30	29	36	35	46	45
17	18	31	32	33	34	47	48
16	13	12	11	54	53	52	49
15	14	9	10	55	56	51	50
2	3	8	7	58	57	62	63
1	4	5	6	59	60	61	64

Itération 3 : numérotation

Solutions logicielles : Renumérotation des données

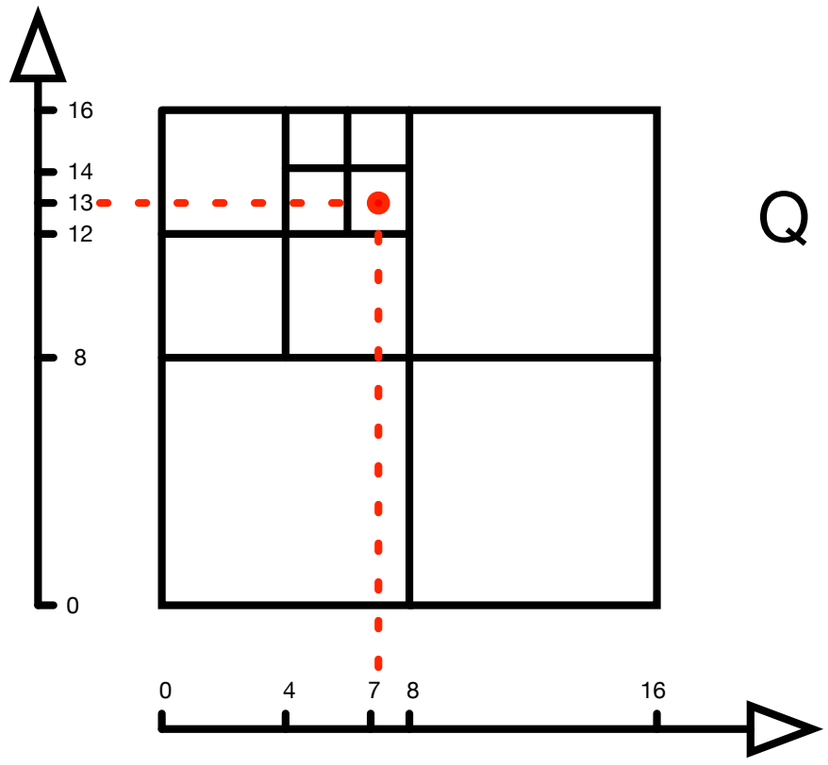
- Renuméroter par une courbe en Z revient à entrelacer les bits des composantes du vecteur de coordonnées.

- On peut remplacer les coordonnées géométriques par les voisins entre triangles ou éléments d'un graphe : on parle alors de renumérotation topologique.



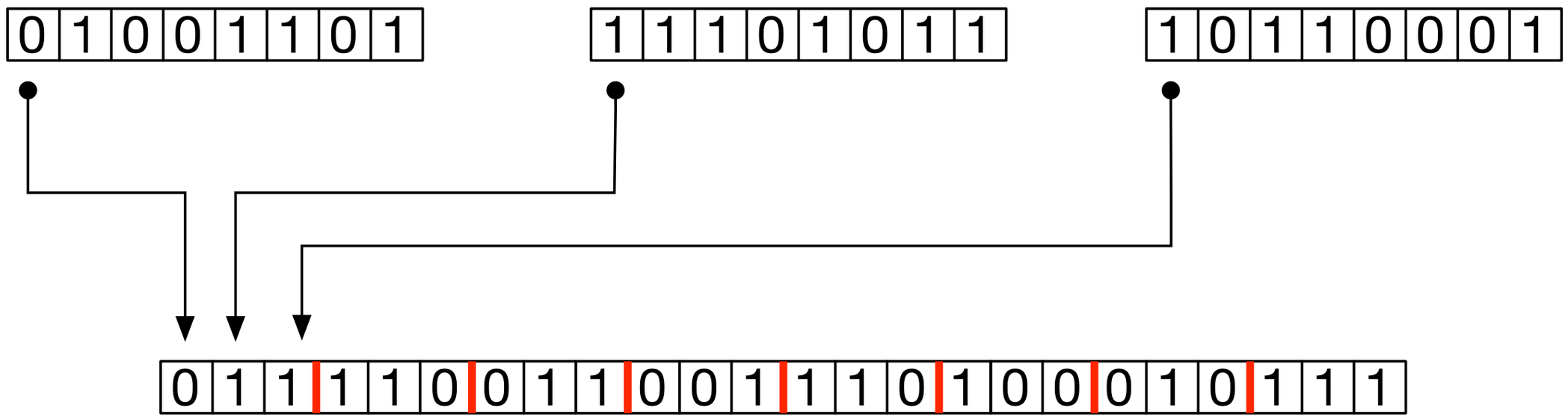
x : 0111 (07)

y : 1101 (13)

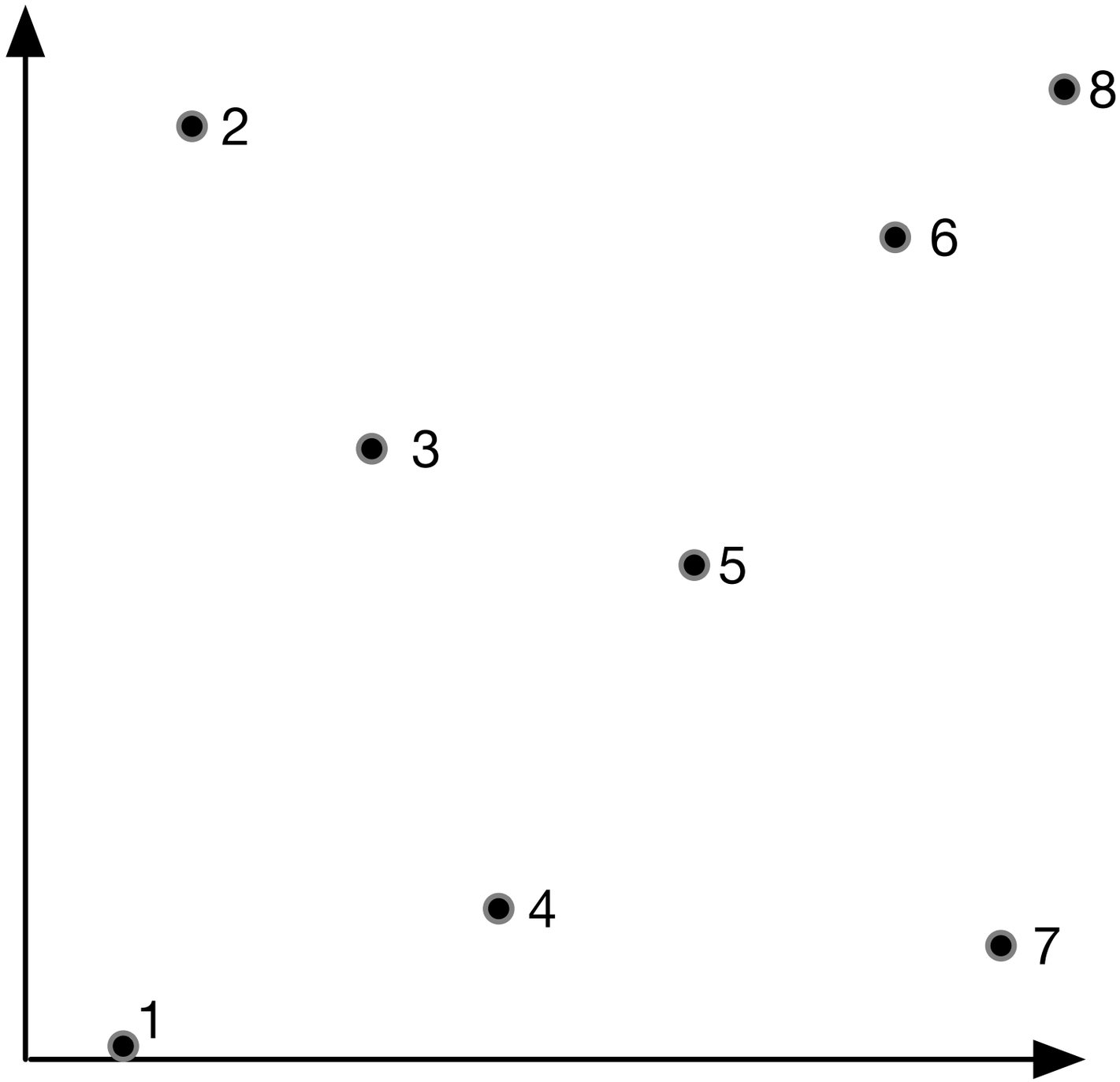


Q : 10.11.01

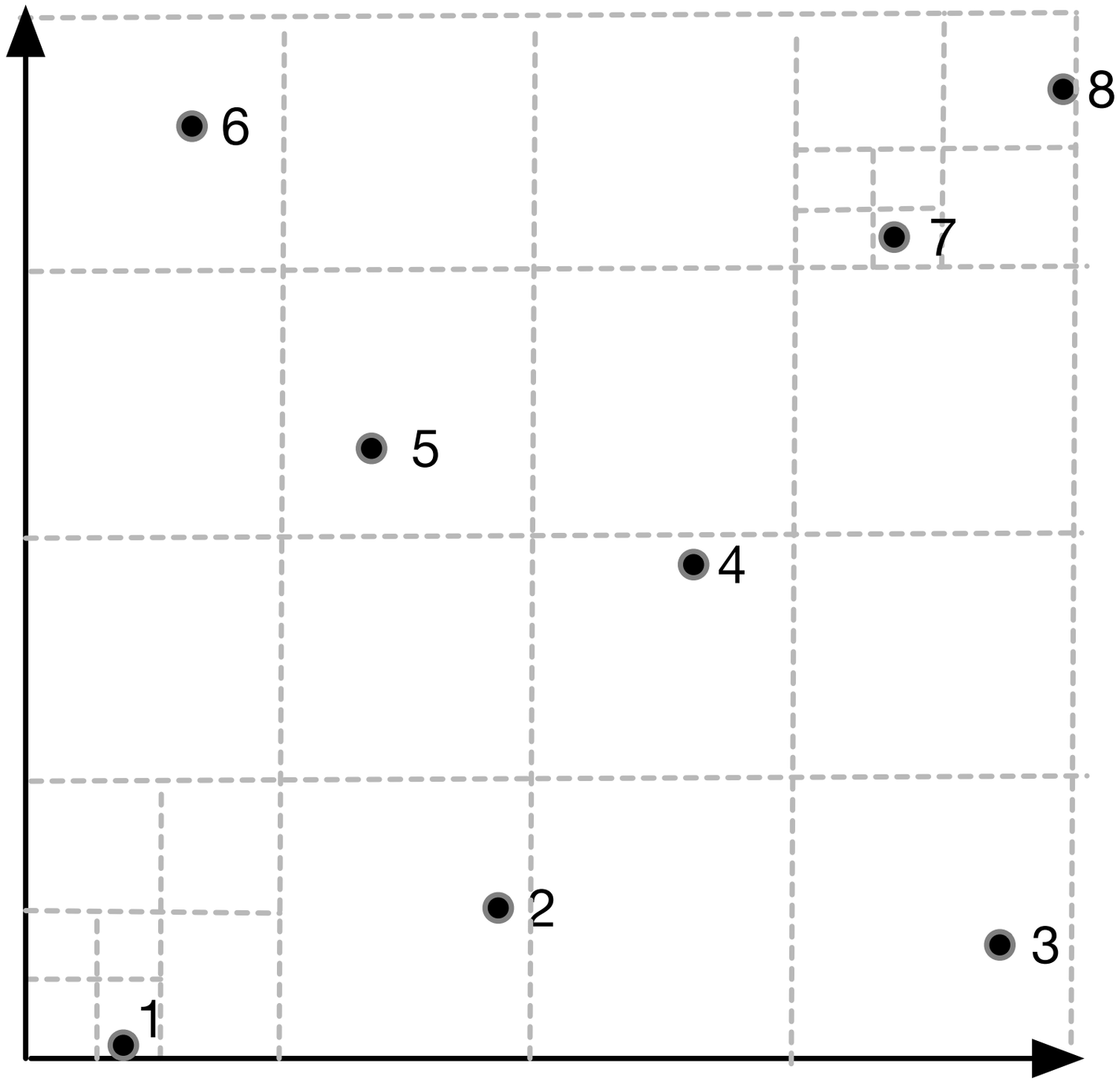
Solutions logicielles : Renumérotation des données

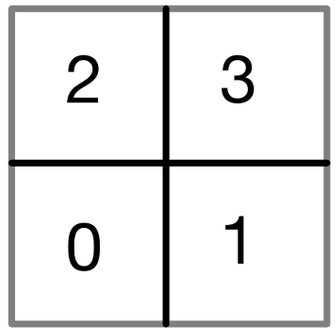


Entrelacement des trois composantes de la position d'un point en un seul scalaire.



Tri des points selon leur coordonnée X





Point 1
x = 1 : 0001
y = 0 : 0000
entrelacées : 00.00.00.01
quadrants : 0 0 0 1
0x64 + 0x16 + 0x4 + 1x1
Q = 1

Point 6
x = 13 : 1101
y = 1100
entrelacées : 11.11.00.01
quadrants : 3301
3x64 + 3x16 + 0x4 + 0x1
Q = 243

Tri des points selon leurs coordonnées X et Y entrelacées

Conclusion :
L'univers a une dimension !

- Toutes les mémoires sont unidimensionnelles, car leur vitesse est due à la lecture parallèle de bloc de données consécutives.

- Optimiser la vitesse d'accès consiste à trouver une fonction de projection pour chaque structure multidimensionnelle sur un espace unidimensionnel.

- Les multiples niveaux de hiérarchie mémoire peuvent être transparents pour le programmeur tant que les accès sont bien linéarisés.

- La vitesse de transfert va continuer à augmenter dans le futur prévisible grâce à l'augmentation de la taille des blocs ce qui rend toutes ces techniques d'optimisation pérennes.