# M3 L3: Bande-passante, stockage et réseau

Remarque: en l'absence de précision, la taille d'un mot est 4 octets

# 1. Bande-passante d'un ordinateur

Vous êtes dans un magasin pour acheter un ordinateur. Vous hésitez entre deux possibilités :

- L'ordinateur A a un processeur à 2 GHz et un disque dur ayant une bande passante de 400 Mo/s. Il a 8 Mo de cache ;
- l'ordinateur B a un processeur à 3 GHz et et un disque dur ayant une bande passante de 12.5 MegaBlocs/s (chaque bloc se compose de 4 mots). Il a 4 Mo de cache.

Que choisissez vous si vous désirez minimiser le temps nécessaire pour :

- a) lire un fichier?
- b) exécuter un programme qui accède souvent à la mémoire ? (Remarquer que les caches de A et B sont du même modèle, mais il y a deux fois plus de blocs dans celui de A que dans celui de B.)

# 2. Bande-passante bis

Vous hésitez entre deux choix d'ordinateurs. Vous demandez plus de détails et découvrez que pour l'ordinateur A, chaque accès au cache prend 1 ns (=1 nanoseconde) et chaque défaut de cache prend 120 ns, tandis que pour l'ordinateur B, chaque accès au cache prend 1.2 ns et chaque défault de cache prend 100 ns. Vous décidez de prendre un exemple d'un programme réel pour comparer les deux ordinateurs. L'algorithme du programme envisagé est le suivant :

```
Algorithme pour programme de test entrée : x entier naturel sortie : y entier naturel y \leftarrow 1
Si \ x = 0 \ \underline{ou} \ x = 1
y \leftarrow 0
s \leftarrow 0, \ i \leftarrow 2
Tant \ que \ s \leq x \ \underline{et} \ y = 1
Si \ i \ divise \ x
y \leftarrow 0
Sinon
i \leftarrow i + 1
s \leftarrow i \times i
Sortir : y
```

Supposons que pour chaque ordinateur la mémoire cache contient 2 blocs et la mémoire en contient 4. Chaque bloc contient 4 mots. L'adresse d'un mot en mémoire est dans l'intervalle [0, 15].

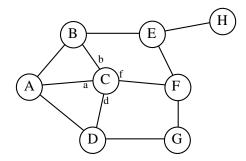
Dans cet exercise on ne compte pas les accès mémoire nécessaires pour lire les instructions. On suppose par ailleurs que le processeur a plus de 4 registres de sorte qu'avec le programme utilisé ici, une fois qu'une variable a été lue dans le cache il n'est pas nécessaire de la relire.

Supposons enfin que x est stocké à l'adresse mémoire 0, y en 1, i en 12, et s à l'adresse 13 (toutes les variables sont initialement en mémoire), et que la valeur de x est initialement 4.

- a) Que fait le programme ci-dessus ?
- b) Explicitez les accès à la mémoire que le programme fait pour lire les 4 variables.
- c) Combien d'accès au cache a-t-on ? Combien de défauts de cache a-t-on ?
- d) Si vous devez choisir l'ordinateur le plus rapide, lequel prendriez-vous sur la base de cet algorithme?

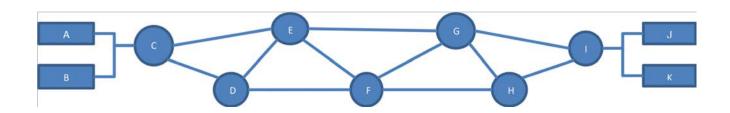
# 3. Routage IP

Considérez le graphe de routeurs Internet suivant :



- a) Simulez l'algorithme de mise à jour de table de routage par « ouï-dire » en partant du nœud A. Notez sur chaque connexion les longueurs de chemin vers le nœud A communiquées par chaque nœud à ses voisins (ne retenez que le plus court). Biffez les longueurs non retenues.
- b) Donnez la table de plus court chemin du nœud C vers tous les autres en indiquant à chaque fois la longueur totale et la première connexion à suivre (cf exemple vers A donné dans la table) :

4 Routage encore: Considérez le graphe de routeurs Internet suivant :



a) Remplissez les informations manquantes dans les extraits des tables de routage suivantes indiquant les chemins vers A et K des routeurs C à I.

		С				D					E					
	(	dest.	di	r. dist	. d	est.	di	ir.	dist	. d	est.	di	r.	dist	:.	
		Α				Α					Α					
		K				K					K					
	F				G		l			Н					I	
dest.	dir.	dist	t.	dest.	dir.	dis	t.	des	st.	dir.	dis	t.	d	est.	dir.	dist.
Α				Α				А	١					Α		
K				K				K	,					K		

b) Comment ces extraits doivent-ils être modifiés si le lien entre E et G est rompu?

С				D		Е			
dest.	dir.	dist.	dest.	dir.	dist.	dest.	dir.	dist.	
Α			Α			Α			
K			K			K			

	F			G			Н			I	
dest.	dir.	dist.									
Α			Α			Α			Α		
К			К			К			К		

c) Comment ces extraits doivent-ils être modifiés si le routeur E tombe en panne?

С				D		E				
dest.	dir.	dist.	dest.	dir.	dist.	dest.	dir.	dist.		
А			Α			Α				
К			K			K				

	F			G			Н			I	
dest.	dir.	dist.									
Α			Α			Α			Α		
K			K			K			K		

# 5 Routage toujours

On considère un réseau dans lequel se trouvent plusieurs routeurs (A, B, C, ..., N). On connaît en partie les tables de routage des routeurs A, F et H qui sont :

	А			F		Н			
dest.	dir.	dist.	dest.	dir.	dist.	dest.	dir.	dist.	
В	С	2	J	0	2	D	I	2	
D	С	5	L	K	2	С	В	2	
N	X	y	М	N	2	F	J	3	

En se basant uniquement sur un strict minimum de liens entre routeurs qui doivent exister selon les tables partielles ci-dessus (i.e. sans en imaginer d'autres non-justifiés par ces tables), quelles sont les valeurs de x et y?

# 6 Encore un peu de routage?

On considère une topologie réseau comprenant 6 routeurs nommés A, B, C, D, E, F. On connaît leurs tables de routages :

Α				В		С				
dest.	dir.	dist.	dest.	dir.	dist.	dest.	dir.	dist.		
С	В	2	D	С	2	F	E	2		
E	F	2	F	Α	2	Α	В	2		
D	В	3	Е	С	2					
	D			E		F				
dest.	dir.	dist.	dest.	dir.	dist.	dest.	dir.	dist.		
В	С	2	D	С	2	В	Α	2		
E	С	2	Α	F	2	С	Е	2		
F	С	3	В	С	2	D	Е	3		
Α	С	3								

Le routeur B tombe en panne et n'est plus utilisable. Tous les autres routeurs sont avertis de cette panne et leur tables de routages sont mises à jour de façon à éviter le routeur B.

Existe-t-il encore une route de A à D après la panne; et si oui quelle est sa longueur?

# 7 **Synthèse musicale** (exercice *optionnel* exploitant des leçons des 3 modules, par J.-C. Chappelier)

Cet exercice porte sur le stockage d'un fichier de musique sur un disque dur.

Rappel et notation utilisée dans cet exercice pour la taille : Les puissances de dix utilisent une seule lettre majuscule comme par exemple :

$$10^3 = K \text{ (kilo)}, 10^6 = M \text{ (méga)}, 10^9 = G \text{ (giga)}$$

les puissances de deux qui sont proches de certaines puissances de dix utilisent la même lettre majuscule suivie par la lettre minuscule 'i'. Voici les exemples proches des puissances de dix indiquées plus haut:

$$2^{10} = 1024 = Ki \text{ (kibi)}, \quad 2^{20} = 1024 * 1024 = Mi \text{ (mébi)}, \quad 2^{30} = 1024 * 1024 * 1024 = Gi \text{ (gibi)}$$

#### 7.1 Besoin de structure

Retrouvez l'amplitude du signal sonore au temps 0.01 s dans l'information suivante stockée sur le disque :

Remarque : vous n'y arriverez pas, il vous manque quelques indications de structuration des données qui sont abordées dans les questions qui suivent.

## 7.2 Echantillonnage du signal (retour sur les leçons II.1–2)

Supposons que l'enregistrement soit celui d'un « la 440» à la qualité CD (échantillonné à 44'100 Hz). Le « la 440 » de l'instrument considéré peut être modélisé comme (nombreuses harmoniques) :

$$X(t) = \sum_{i=1}^{5} 0.8^{i-1} \, \sin(2\pi \, i \, f \, t)$$
 avec  $f$  =440 Hz.

Ce signal peut-il être reconstitué exactement à partir des échantillons du CD? Justifiez pourquoi. Sinon, que devrait-on faire?

## 7.3 Codage du signal (retour sur les leçons II.3–4)

L'amplitude du signal à chaque échantillon est stockée sur 16 bits. Combien de niveaux distincts d'amplitudes dispose-t-on, sachant que cette amplitude peut avoir un signe négatif ?

Pour le stockage sur disque, on aimerait pouvoir comprimer de 50% le signal échantillonné avec une compression sans perte et sans préfixe. Pour le savoir, on applique le théorème de Shannon ; sachant que l'entropie des valeurs de l'amplitude est de 10.5 bit. Est-ce possible de comprimer (sans perte et sans préfixe) à 50%?

→ Pour répondre à cette question, utiliser le fait que l'entropie représente la performance maximale, exprimée en nombre de bit par échantillon d'amplitude, qu'on puisse espérer pour un codage donné.

On utilise finalement le code de Huffman pour représenter les valeurs échantillonnées et quantifiées. A partie de la relation entre l'entropie et la performance du codage de Huffman, quel intervalle de taux de compression peut-on espérer au mieux (ordre de grandeur)?

## 7.4 Stockage sur disque : contenu du fichier (leçons I.1 et III.3)

On suppose maintenant que notre code de Huffman a une longueur moyenne de 11 bits/lettre.

Le disque utilise des **blocs de 4 Kio**. Nous avons enregistré **2 s** de musique.

Combien de blocs (environ) utilise notre fichier de 2 secondes ?

Nous avons un disque de 512 Gio : combien de blocs contient ce disque? Combiens de bits sont nécessaires pour l'adresse des blocs du disque ?

On utilise finalement des adresses de blocs sur 32 bits.

Les adresses des blocs de notre fichier sont stockées dans une table qui commence comme cela :

Décoder la valeur binaire présente dans les deux premières lignes ci-dessus (entier non-signé) :

La valeur codée sur la première ligne est la taille de la table, c'est à dire le nombre d'adresses de blocs que contient la table (nombre de bloc qu'on a trouvé à la question précédente pour nos 2 secondes de musique).

La valeur suivante, sur la seconde ligne, est donc l'adresse du premier bloc.

Si l'on suppose que tous les mots de code utilisés par le code de Hufmann précédent ont en moyenne 11 bits (approximation), quelle est l'adresse du bloc correspondant à l'amplitude du signal sonore au temps 0.01 s?

→ Pour cela, trouver d'abord à quel numéro d'échantillon ce temps correspond puis l'espace nécessaire pour coder tous les échantillons jusqu'à celui-là. Enfin, il suffit de comparer ce nombre à la taille du bloc (4 Kio).

### 7.5 Stockage sur disque : nom absolu du fichier (leçon III.3)

Notre fichier de musique est lui-même stocké dans un répertoire, par exemple dans /home/muche/Musique.

Supposons que du point de vue du disque, les répertoires sont simplement des fichiers particuliers contenant les tables des fichiers qu'ils contiennent :

 le répertoire racine (/) est un fichier contenant les tables de ses fichiers et répertoires, dont celle du répertoire home;

- le répertoire home est un fichier contenant les tables de ses fichiers et répertoires, dont celle du répertoire muche ...
- le répertoire Musique est un fichier contenant les tables de ses fichiers et répertoires, dont celle de notre fichier (donnée plus haut).

Si l'on suppose que notre fichier est le seul dans le répertoire **Musique**, on va y trouver seulement la *table* donnant l'adresse des blocs de ce fichier. Combien de blocs utilise alors <u>ce répertoire</u> Musique sachant que la table commence par indiquer le nombre de blocs occupés par le fichier et qu'une adresse occupe 32 bits pour cet exercice ?

En supposant que les blocs du répertoire Musique sont stockés à partir de l'adresse **42**, et en supposant que le répertoire **muche** ne contient que le répertoire Musique, quel est le début du fichier représentant le répertoire muche (donner le contenu binaire de sa table des adresses des blocs du répertoire muche) ?

## 7.6 Décodage du « message » de départ

#### Sachant

 que les adresses indiquées ci-dessous sont au format : « adresses de bloc.adresse de mot », avec un bloc de taille 4Kio et des mots de 32 bits

— que le disque contient les informations suivantes :

— que le code de Huffman utilisé est le suivant (les quelques valeurs ci-dessous nous suffisent) :

valeur sur 16 bits	valeur sur 16 bits	Code de		
(décimale)	(binaire)	Huffman		
 -26922	1110100100101010	01010101010101		
0	000000000000000000000000000000000000000	11001100110		
 5929	0001011100101001	10101010		
 11056	0010101100110000	0001110001		
26922	0110100100101010	11111111111111		

Quelle est la valeur décimale de l'amplitude représentée sur 16 bits pour notre signal au temps 2/44100 s sachant que la mesure commence au temps 0s ?

 $\rightarrow$  pour cela, utiliser l'adresse trouvée à la question 7.4 et décoder les codes de huffman jusqu'à celui correspondant au  $3^{ième}$  échantillon.

L'amplitude maximum sur 16 bits correspond en fait à la valeur 4 ; sachant cela, on peut vérifier qu'on obtient bien la valeur de la somme donnée en 7.2 pour le temps 2/44100.