

Rapport : La dissémination de données.

Introduction

L'approche pseudo interactive

Avec l'usage croissant des télécommunications sans fils, en particulier des téléphones mobiles, on assiste à une nouvelle donne dans la diffusion d'information : des serveurs sont amenés à fournir de l'information à un parc d'utilisateurs très important alors que le retour d'information est faible (ne serait-ce que parce que le débit d'envoi est bien moins important que le débit reçu). On parle alors d'environnements asymétriques. Dans ce type d'environnement, l'approche interactive atteint ses limites : les temps de diffusion sont trop longs et surtout, le serveur ne peut plus gérer l'afflux de clients trop important. Pour pallier à ces inconvénients, une nouvelle approche a fait jour : la méthode pseudo-interactive. Le serveur diffuse les messages dans un ou plusieurs canaux réservés et le client se connecte à ces canaux et attend que le message qu'il a demandé soit diffusé. Le client a seulement l'illusion de l'interactivité, d'où le nom.

Cette méthode appliquée à l'Internet est l'une des plus prometteuses pour réduire la saturation du réseau, liée à l'augmentation exponentielle du nombre d'utilisateurs. En effet, seul un petit nombre des informations disponibles est demandé par la plupart des requêtes. Ainsi, pour le serveur cs-www.bu.edu, 0.5% des pages sont demandées par 61% des requêtes, et 10% par 91% des requêtes. Utiliser l'approche pseudo-interactive pour diffuser le petit nombre de messages les plus populaires permettrait à la fois de gagner de la bande passante et de libérer le serveur du traitement d'un grand nombre de requêtes.

Approche du TIPE

Dans ce rapport, nous nous contenterons de l'étude des messages ayant des longueurs uniformes. En effet lorsque les messages sont de longueurs différentes, le problème est nettement plus difficile. En particulier, des ordonnancements optimaux peuvent présenter des trous (dans lesquels aucun message n'est diffusé), et ce même s'il n'y a pas de coûts de diffusion.

Dans la première partie, nous commençons par établir le coût d'un ordonnancement, ce qui nous permettra de trouver un minorant du coût de tout ordonnancement périodique (il s'agit du minorant d'Ammar et Wong étendu au cas de plusieurs canaux avec coûts de diffusion). Le minorant nous indiquant les fréquences optimales, il nous permettra de décrire plusieurs algorithmes de dissémination de données.

Dans la deuxième partie, nous étudions la performance en pratique des algorithmes donnés ci-dessus. En fait, nous ne nous intéressons pas à la mise en place pratique de la dissémination de données mais nous évaluons l'efficacité intrinsèque de ces algorithmes suivant les différentes distributions de popularité et d'autres contraintes. C'est pourquoi nous avons écrit un programme qui simule un serveur ayant à traiter les requêtes de clients se loguant en un temps continu. Le serveur a le choix entre l'approche interactive ou l'un des algorithmes issus de l'approche pseudo-interactive. La simulation retourne ensuite le temps d'attente moyen des clients.

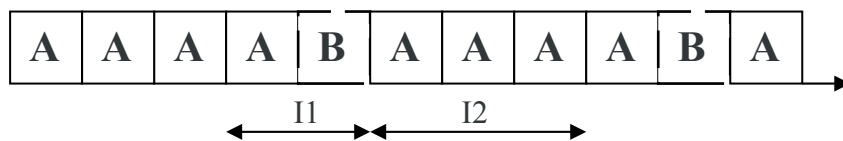
1^{ère} partie : Présentation des résultats théoriques.

1.1-Exemple

Nous commençons par présenter un exemple, afin de clarifier les notions. Nous avons deux messages A et B de même longueur à diffuser sur un canal, sachant que A est quatre fois plus populaire que B. Quel ordonnancement choisir pour minimiser le temps d'attente des clients ? A priori, l'ordonnancement optimal diffuse plus souvent A que B... Examinons deux cas.

Première idée

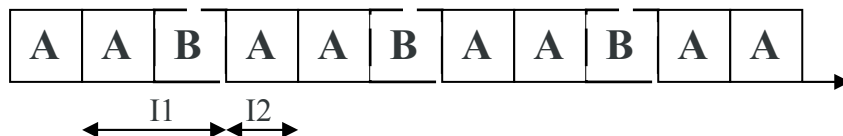
Diffuser A quatre fois plus que B.



Une requête pour B attend $5/2 = 2.5s$ plus le temps de téléchargement de B soit au total 3.5 secondes. Une requête pour A tombe dans I1 avec une probabilité de $2/5$ et attend $2/2 + 1 = 2s$ et tombe dans I2 avec une probabilité de $3/5$ et attend $1/2 + 1 = 1.5s$ soit au total $1.7s$. Le temps d'attente moyen est donc de $0.8 \times 1.7 + 0.2 \times 3.5 = 2.06s$.

Seconde idée

Diffuser A $2 = \sqrt{4}$ fois plus que B.



On a de même un temps d'attente pour B de $1 + 1.5 = 2.5s$ et un temps d'attente pour A de $1 + 2/3 \times 1 + 1/3 \times 0.5 = 1.83s$. Cela nous donne un temps d'attente moyen de $0.8 \times 1.83 + 0.2 \times 2.5 = 1.97s$. Nous verrons que c'est le temps d'attente optimal : il faut diffuser les messages proportionnellement à la racine carré de leur popularité.

1.2-Calcul du temps d'attente moyen

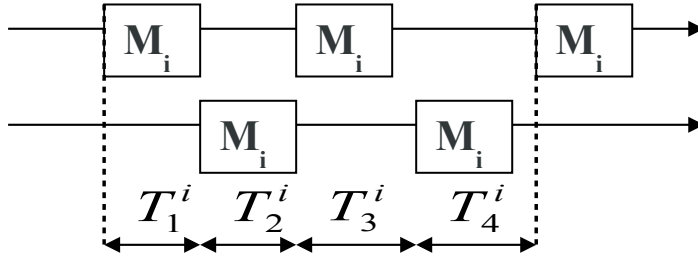
Données du problème

On a un ensemble de m messages $\{M_1, \dots, M_i, \dots, M_m\}$ de longueurs uniformes à diffuser sur W canaux. Chaque message M_i a une popularité p_i et un coût de diffusion c_i . On définit un ordonnancement par un ensemble de W suites $\{s_1, \dots, s_w\}$ de messages. Si $s_i(n) = M_j$, le message M_j est diffusé au temps n dans le $i^{\text{ème}}$ canal. Si $s_i(n) = X$, aucun message n'est diffusé dans le canal i au temps n . L'unité de temps est ici le temps de téléchargement d'un message.

On définit le temps de service TS d'un ordonnancement par le temps d'attente moyen. Si $TS(M_i)$ est le temps d'attente moyen pour une requête du message M_i , alors $TS = \sum_{i=1}^m TS(M_i)$. On définit le coût de diffusion C_DIFF par le coût moyen de diffusion des messages. On peut alors définir le coût d'un ordonnancement par $COUT = TS + C_DIFF$.

On se restreint aux ordonnancements périodiques (en fait, on peut montrer qu'il existe toujours un ordonnancement périodique optimal, donc cette hypothèse ne nuit pas à la

généralité). Nous allons introduire quelques notations. Considérons un ordonnancement périodique de période T.



On définit :

- n_i est le nombre de diffusions de M_i durant la période.
- T_1^i est l'intervalle de temps qui sépare le début de la période de la première diffusion de M_i . $T_j^i, 2 \leq j \leq n_i$ est l'intervalle de temps qui sépare les débuts de la $(j-1)^{\text{ième}}$ et la $j^{\text{ième}}$ diffusion de M_i .

Nous allons exprimer le coût d'un ordonnancement en fonction des n_i et des T_j^i :

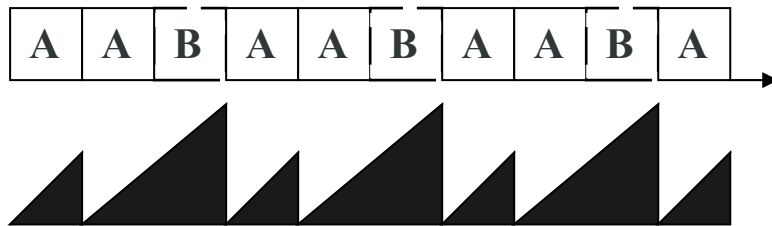
Lemme 1.1 - Soit S un ordonnancement périodique de période T. Alors le coût de S est

$$1 + \frac{1}{T} \sum_{i=1}^m \left(p_i \sum_{j=1}^{n_i} \frac{(T_j^i)^2}{2} + n_i c_i \right)$$

Preuve :

Calculons $TS(M_i)$: une requête pour M_i a une probabilité de T_j^i / T de tomber dans l'intervalle T_j^i et attend en moyenne le début du message $T_j^i / 2$ secondes. On obtient donc $TS(M_i) = 1 + \sum_{j=1}^{n_i} (T_j^i)^2 / (2T)$ en ajoutant le temps de téléchargement. De plus, on a directement $C_DIFF(M_i) = n_i c_i / T$. La formule en découle, puisque $COUT(S) = \sum_{i=1}^m p_i [TS(M_i) + C_DIFF(M_i)]$ ☺

On a alors une représentation visuelle simple de $TS(M_i)$ comme aire moyenne de triangles isocèles :



On voit que $TS(A) = 1 + \frac{1}{3} \left(\frac{1^2}{2} + \frac{2^2}{2} \right) = 1.83s$.

1.3-Calcul du minorant

Le résultat précédent va nous permettre de calculer un minorant du coût de tout ordonnancement périodique.

Théorème 1.1 :

$$\text{Soit } Min1: \min \left(1 + \sum_{i=1}^m \left(\frac{p_i \tau_i}{2} + \frac{c_i}{\tau_i} \right) \right) \text{ sur le domaine } \begin{cases} \forall i, \tau_i \geq 1 \\ \sum_{i=1}^m \frac{1}{\tau_i} \leq W \end{cases}$$

$Min1$ est un minorant de tout ordonnancement périodique et est atteint pour un unique τ^* . Un ordonnancement périodique S a un coût égal à ce minorant ssi il diffuse M_i tous les τ_i^* .

Preuve :

Nous allons d'abord minimiser $TS(M_i)$ (en relaxant la contrainte de non recouvrement des messages). On a $TS(M_i) = 1 + \sum_{j=1}^{n_i} (T_j^i)^2 / (2T)$ avec $\sum_{j=1}^{n_i} T_j^i = T$. Comme la fonction $x \rightarrow x^2$ est strictement convexe, $TS(M_i)$ est minimisé ssi les T_j^i ont même longueur : T/n_i .

Posons $\tau_i = T/n_i$. Alors $TS(M_i) \geq 1 + \tau_i/2$. Donc $TS \geq 1 + \sum_{i=1}^m \left(\frac{p_i \tau_i}{2} + \frac{c_i}{\tau_i} \right)$ puisque $\frac{c_i n_i}{T} = \frac{c_i}{\tau_i}$.

De plus $n_i \leq T$ donc $\forall i, \tau_i \geq 1$ et $\sum_{i=1}^m n_i \leq WT$ donc $\sum_{i=1}^m 1/\tau_i \leq W$, ce qui nous donne le domaine de minimisation. Enfin, la fonction objectif est continue sur un domaine fermé et tend vers l'infini quand $\|\tau\|$ tend vers l'infini donc elle admet un minimum (les fermés bornés sont compacts en dimension finie). De plus elle est strictement convexe sur le domaine convexe défini par les contraintes donc ce minimum est unique. Le reste en découle alors facilement.

Il ne reste plus qu'à calculer τ^* , c'est l'objet du théorème qui suit.

Théorème 1.2 :

Soit $\tau_i' = \sqrt{\frac{2c_i + \lambda}{p_i}}$ avec λ défini par :

- si $\sum_{i=1}^m \sqrt{p_i / (2c_i)} \leq W$ alors $\lambda=0$
- sinon λ est l'unique solution positive de $\sum_{i=1}^m \sqrt{p_i / (2c_i + \lambda)} = W$

Alors pour tout $1 \leq i \leq m$ tel que $\tau_i' < 1$, alors $\tau_i^* = 1$. On itère alors les calculs (en enlevant les messages et le nombre de canaux correspondants) jusqu'à ce que pour tout i , $\tau_i' \geq 1$. Alors $\tau_i^* = \tau_i'$.

On obtient bien une fréquence de diffusion proportionnelle à la racine carrée de la popularité quand les coûts de diffusion sont nuls.

1.4-Présentation des algorithmes

Le minorant nous indique des fréquences de diffusion de $1/(W\tau_i)$. Ceci nous amène à présenter quelques algorithmes. En fait, trouver l'optimal étant NP-dur, les algorithmes que

nous présentons sont des α -approximations. Il se peut que $\sum_{i=1}^m 1/\tau_i^*$ soit plus petit que W . Dans ce cas on rajoute un message fantôme M_0 avec $p_0=c_0=0$ et $1/\tau_0^* = W - \sum_{i=1}^m 1/\tau_i^*$.

Algorithme randomisé

On diffuse le message M_i avec une probabilité $s_i = 1/(W\tau_i^*)$. On remarque que l'on a $\sum_{i=0}^m s_i = 1$ et qu'avec une probabilité de $(1 - \sum_{i=1}^m s_i)$ on ne diffuse rien.

Alors l'espérance de $TS(M_i)$ est de $1 + 1/2 + \sum_{n \geq 1} (1 - Ws_i)^n = 1/2 + \tau_i^*$ donc l'espérance de TS est $1/2 + \sum_{i=1}^m p_i \tau_i^*$. De même on a l'espérance de C_DIFF égale à $\sum_{i=1}^m c_i s_i W = \sum_{i=1}^m c_i / \tau_i^*$. Donc l'espérance du coût de cet algorithme est plus petit que $2 * \text{Min}_1$. L'algorithme est donc une 2-approximation.

Glouton

L'idée est de diffuser le message dont l'attente cumulée est maximum. Si on note σ_i le temps qui s'est écoulé depuis le début de la dernière diffusion du message M_i , le glouton diffuse le message M_i tel que $p_i \sigma_i \tau_i^* - c_i$ soit maximal.

On peut démontrer que cet algorithme est la dérandomisation gloutonne de l'algorithme précédent. Il donnera ainsi toujours de meilleurs résultats que celui-ci. C'est donc une 2-approximation.

Nombre d'or

L'idée est de construire un ordonnancement périodique qui respecte les proportions $1/W\tau_i^*$ des messages définis par le minorant. Pour cela on va se servir d'une propriété du nombre d'or :

$\varphi = \frac{1+\sqrt{5}}{2}$ est l'irrationnel dont les multiples modulo 1 découpent $[0,1]$ le plus uniformément possible.

Soit T la longueur de l'ordonnancement périodique à construire, $n_i = \lfloor T/\tau_i^* \rfloor$ le nombre de diffusion de M_i pendant cet intervalle. On attribue à M_i un ensemble $X_i = \{(n_0 + n_{i-1} + j)\varphi \bmod 1, 1 \leq j \leq n_i\}$ de n_i position puis on trie $X_1 \cup \dots \cup X_m$ (dans l'ordre croissant). On diffuse alors dans le $n^{\text{ième}}$ créneau le message M_i tel que la $n^{\text{ième}}$ position appartienne à X_i .

L'algorithme est une 9/8 approximation (lorsqu'il n'y a pas de coût de diffusion).

2^{ème} partie : Résultats de la simulation.

2.1-Utilité d'une simulation

Nous avons donc plusieurs algorithmes à notre disposition. Il reste à déterminer lequel utiliser dans quelle situation. Par ailleurs, lorsqu'on a le choix d'autres stratégies (on songe ici surtout à Internet), il nous faut savoir quand opter pour la dissémination de données se révélera payant. Pour cela, nous avons besoin d'une simulation qui, en fonction du nombre de messages, de leur popularité, du nombre de clients à servir, du nombre de canaux à

disposition, nous donne le temps d'attente moyen par clients. C'est cela que réalise notre programme, dont le code source est donné en annexe.

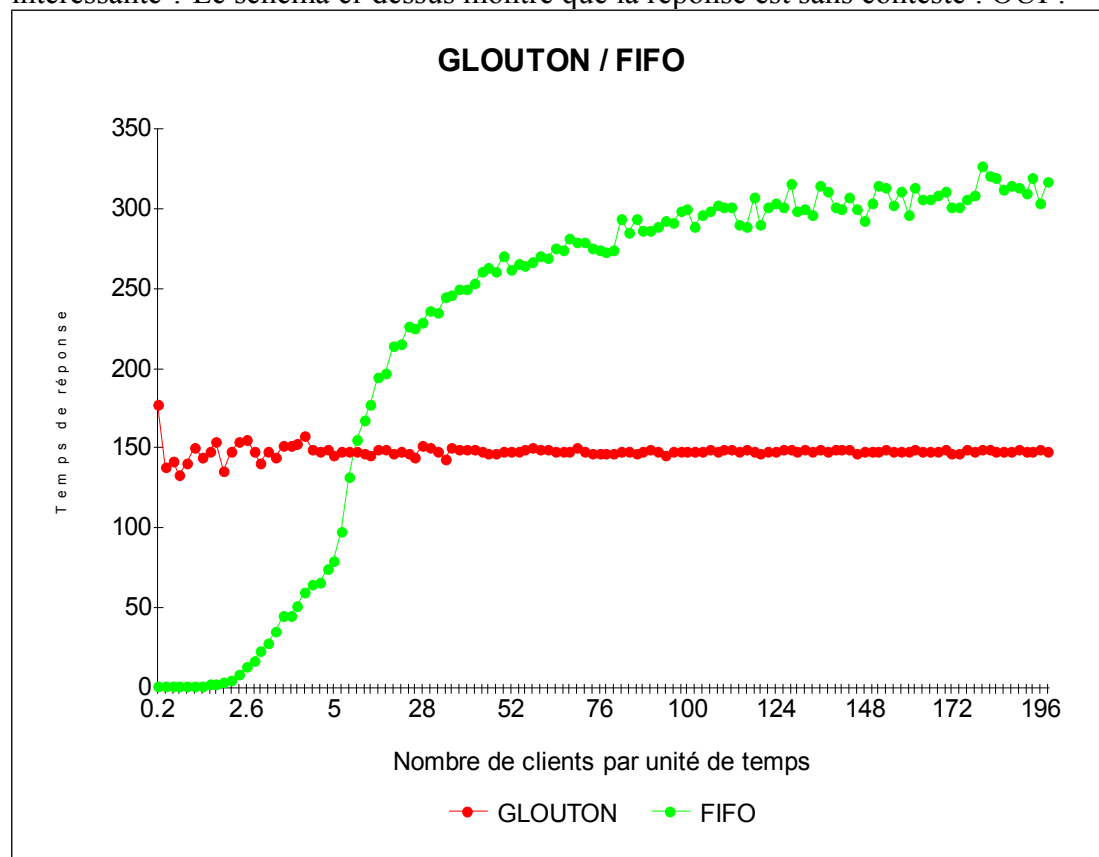
La principale difficulté de ce programme ne réside pas en la programmation des algorithmes (pour ceux-ci le point délicat est le calcul des τ_i^* , c'est-à-dire de λ , mais pour cela nous utilisons la méthode des tangentes de Newton), mais bien en la simulation proprement dite. En effet, la méthode retenue consiste à considérer que les clients se loguent en un temps continu pendant un certain intervalle de temps tandis que le serveur diffuse des messages sur les canaux à disposition, le programme renvoyant le temps d'attente moyen par clients. Pour obtenir des résultats cohérents, il importe que la simulation se déroule sur un intervalle de temps assez long, ce qui implique d'optimiser au maximum cette partie du programme. On trouvera dans le code source du programme une explication plus détaillée des difficultés rencontrées et des choix effectués.

Les résultats de la simulation que nous présentons ont un double but : montrer l'intérêt que peut avoir cette approche, et, dans un deuxième temps, savoir quels algorithmes il vaut mieux utiliser dans quel cas.

2.2-Principales données recueillies par la simulation

Comparaison entre l'approche pseudo-interactive et l'approche usuelle

C'est bien sûr la principale interrogation : est-ce que l'approche pseudo-interactive est intéressante ? Le schéma ci-dessus montre que la réponse est sans conteste : OUI !



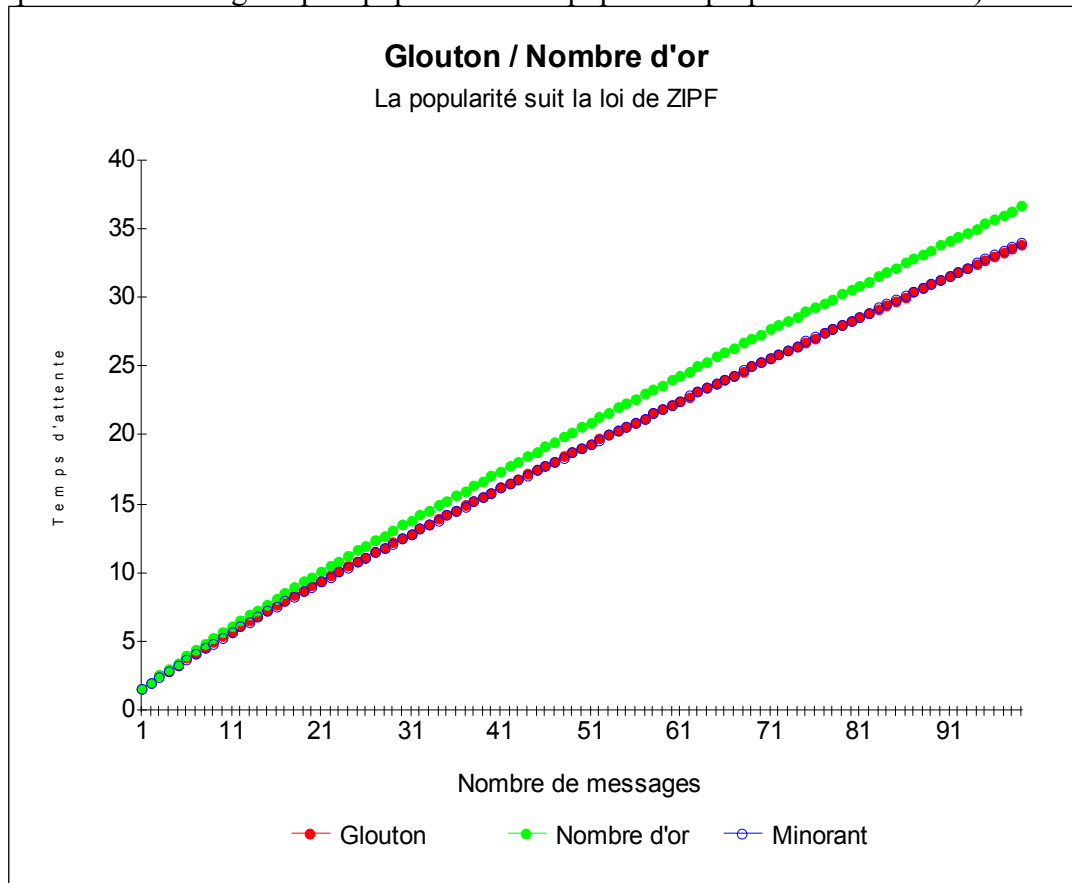
La simulation s'est effectuée sur un ensemble de 1000 messages à diffuser sur un canal. On remarque que le temps d'attente moyen pour l'approche pseudo-interactive reste constant quelque soit le nombre de visiteurs tandis que pour l'approche interactive habituelle, si le temps d'attente est quasiment nul lorsqu'il y a peu de clients connectés au site, il croît très

vite avec le nombre de clients pour se stabiliser à une valeur d'attente bien supérieure à celle du glouton. (La forme en S s'expliquant qu'à la fin, tous les messages sont dans la liste d'attente, ce qui fait que l'algorithme par FIFO diffuse les messages cycliquement et donc tend vers un temps d'attente de $1000/2=500$).

On voit ici tout l'intérêt de l'approche pseudo-interactive : puisque le temps d'attente ne varie pas avec le nombre de clients, on pourra utiliser cette méthode pour diffuser les messages les plus populaires, en utilisant la méthode par FIFO usuelle seulement pour les messages les moins demandés.

Comparaison entre le glouton et l'algorithme du nombre d'or

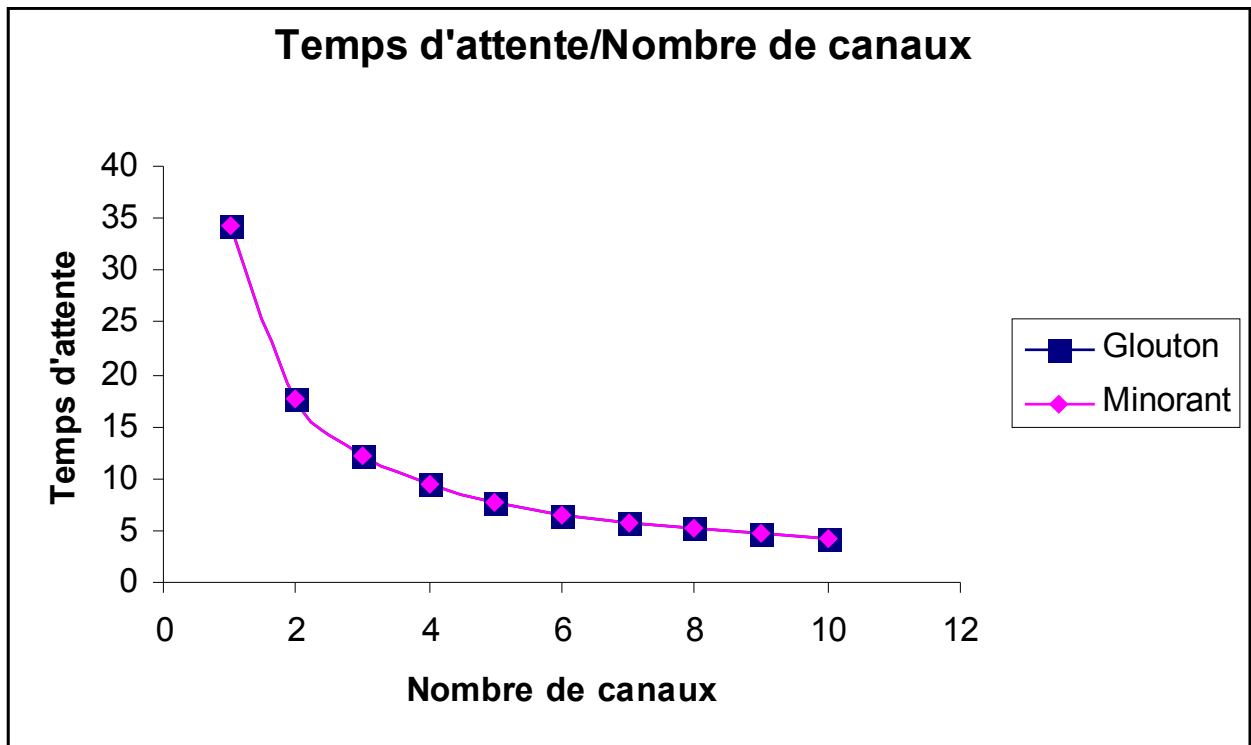
On a vu l'utilité de la dissémination de données par rapport à une approche interactive. Reste à savoir quel algorithme de dissémination de données utiliser. On sait que le glouton donne toujours de meilleurs résultats que l'algorithme randomisé. Reste à comparer le glouton avec celui du nombre d'or. Or l'algorithme du nombre d'or est une $9/8$ -approximation tandis que le glouton n'est qu'une 2-approximation. On a donc une meilleure borne pour l'algorithme du nombre d'or, mais cela ne veut pas dire qu'il donnera de meilleurs résultats en pratique. Voici le résultat de la simulation lorsque la popularité des messages suit la loi de ZIPF (c'est-à-dire que le $n^{\text{ième}}$ message le plus populaire a une popularité proportionnelle à $1/n$) :



La simulation s'est effectuée sur 1 canal, la période de l'algorithme du nombre d'or étant de $1000 \times$ le nombre de messages. On constate que lorsque la popularité suit la loi de ZIPF, le glouton donne de meilleurs résultats, très proches de l'optimal. Or il se trouve que cette loi modélise bien le comportement des internautes. Donc pour une application de la dissémination de données à Internet, on pourra préférer le glouton.

Variation du temps d'attente en fonction du nombre de canaux :

Ici on s'intéresse à la variation du temps de service en fonction du nombre de canaux : plus il y a de canaux de diffusion, plus on peut diffuser rapidement des messages. Dans quelle mesure cela influe-t-il sur le temps de service?



On voit que le temps d'attente « réel » suit la loi théorique en K/W (où W représente le nombre de canal).

L'ajout d'un canal permet de diminuer le coût d'un facteur d'au plus $\frac{W}{W+1}$. Ainsi il vaut mieux utiliser un meilleur algorithme que de chercher à augmenter le nombre de canal pour augmenter la rapidité.

2.3- Synthèse

Rappelons les avantages de la dissémination de données. D'abord, le serveur diffuse les informations sans avoir à traiter les requêtes utilisateurs, ce qui permet de libérer des ressources. Ensuite, le principe même de la dissémination de données est de diffuser un certain nombre de messages populaires sur un petit nombre de canaux, ce qui permet de libérer de la bande passante. Enfin, et surtout, le temps de service ne dépend que de l'ordonnancement choisi et pas du nombre de clients. Ceci permet de garantir un temps de service constant quel que soit la situation, et en particulier cela permet d'éviter les si frustrants « Denial of Service »...

La simulation montre que le temps de service est tout à fait satisfaisant et donne de (bien) meilleurs résultats que la méthode habituelle dès que le serveur a à traiter un certain nombre de requêtes. On pourrait donc envisager un système hybride, utilisant les algorithmes de dissémination de données pour les messages les plus populaires, et la méthode habituelle pour les autres.

Conclusion

Nous espérons que malgré la brièveté de cette introduction à la dissémination de données, son utilité ne fait plus aucun doute. Bien sûr, notre programme nous a permis d'obtenir de nombreux autres résultats, et nous avons en réalité étudié d'autres algorithmes que les « classiques » qui sont présentés ici. En particulier, M. Schabanel, chercheur au CNRS, nous a conseillé d'étudier un algorithme dynamique, qui s'adapte aux variations de centre d'intérêt des utilisateurs au cours de la journée. Cet algorithme donne effectivement de très bon résultats : il est aussi rapide que le FIFO lorsqu'il y a peu de clients, et il est toujours plus rapide que le glouton, même lorsque le nombre de clients est important. Nous espérons pouvoir en parler lors de notre présentation...

Toutefois des problèmes pratiques peuvent survenir lorsqu'on souhaite mettre en application la dissémination de données (outre le fait qu'on s'est limité ici à des messages de même longueur) : on suppose connue la popularité des pages demandées par les clients, alors que cette dernière peut changer au cours du temps. Malgré cela, par son grand potentiel qui n'est pas encore totalement exploré, ses nombreuses utilisations possibles, nous pouvons sans craintes affirmer que la dissémination de données sera au cœur du futur de la diffusion d'information.