

Vers un système hybride pour l'optimisation du trafic réseau et de la bande passante internet

Amir Hajjam, Abder Koukam, Lyamine Bouhafs

*Université de Technologie de Belfort-Montbéliard
Équipe Systèmes Multi-Agents
Laboratoire Systèmes et Transports
90010 Belfort cedex, France*

amir.hajjam@utbm.fr

abder.koukam@utbm.fr

lyamine.bouhafs@utbm.fr

Résumé: Le réseau Internet est de plus en plus utilisé par les particuliers, les entreprises, les gouvernements ainsi que d'autres organisations diverses. Les échanges de données qui en résultent sont de plus en plus massifs et impliquent des charges des serveurs élevées, mais surtout des lignes d'interconnexion de plus en plus saturées. Beaucoup de protocoles sont utilisés sur Internet, mais celui qui permet l'existence du web, la toile hypertexte, est celui qui prédomine : HTTP. Les données qui transitent par ce protocole (les pages web, les images et d'autres contenus divers) se prêtent particulièrement bien à la mise en cache. Dans cet article, nous montrons qu'il est possible d'améliorer substantiellement la performance des systèmes de mise en cache, et de diminuer le niveau de bande passante maximale nécessaire dans un réseau en utilisant des algorithmes avancés.

Mots clés: Algorithmes avancés, Internet, Optimisation, Trafic réseau.

1 Introduction

Le réseau Internet est de plus en plus utilisé par les particuliers, les entreprises, les gouvernements ainsi que d'autres organisations diverses. Les échanges de données qui en résultent sont de plus en plus massifs et impliquent des charges des serveurs élevées, mais surtout des lignes d'interconnexion de plus en plus saturées.

Le niveau atteint par ces échanges, majoritairement de nature consultative, impose la mise en place de systèmes de réplique qui conservent les données à des endroits physiquement plus proches de leurs utilisateurs finaux. Ainsi, il est possible de mettre en jeu moins de connexions que si ces données avaient récupérées depuis le serveur d'origine.

Beaucoup de protocoles sont utilisés sur Internet, mais celui qui permet l'existence du Web, la toile hypertexte, est celui qui prédomine : HTTP. Les données qui transitent par ce protocole (les pages web, les images et d'autres contenus divers) se prêtent particulièrement bien à la mise en cache.

Cette technique a largement fait ses preuves dans d'autres domaines, notamment dans le matériel électronique et informatique (microprocesseurs, par exemple) où les caches assurent des fonctions d'interface entre des systèmes lents et des systèmes

rapides, ce qui est le cas dans le cas de réseaux locaux connectés à Internet par des liaisons à plus faible débit.

De nos jours, malgré l'existence réelle de solutions de mise en cache et de leurs évolutions depuis une dizaine d'années, leur relative simplicité est de moins en moins adaptée à la nature dynamique des contenus du réseau et aux comportements des utilisateurs. L'irrégularité et l'importance croissante du trafic est parfois telle qu'il faut envisager d'augmenter la performance des caches.

2 La mise en cache

2.1 Présentation

Les serveurs mandataires ou proxy couplés à des services de mise en cache des documents sont des éléments essentiels du réseau Internet, du moins pour les données transférées par le protocole HTTP, ce qui représente la majeure partie du trafic. Sans eux les requêtes et le transfert de documents devraient toujours traverser l'intégralité du chemin entre le client et le serveur d'origine du document. Les besoins en bande passante seraient alors plusieurs fois plus élevés que ce qui est utilisé actuellement, et les temps de réponse si élevés qu'une navigation agréable serait illusoire.

Ces services de mise en cache sont étudiés depuis les débuts de l'Internet (Abrams & al., 1995). Ces études ont permis d'en confirmer l'efficacité et d'améliorer le protocole HTTP pour qu'il prenne en compte l'existence de systèmes de mise en cache (Network Working Group, 1999). Si les premières expériences avançaient des chiffres d'utilisation des caches allant de 30 à 50%, il n'est pas rare aujourd'hui d'évoluer sur des fourchettes allant de 40 à 50%.

En essence, ces systèmes permettent de rapprocher les données de leurs utilisateurs, économisant ainsi de la bande passante et améliorant les temps de réponse en permettant aux requêtes de parcourir un chemin beaucoup plus court que si elles avaient été relayées jusqu'au serveur d'origine.

Le principe de fonctionnement de ces serveurs est simple : placés sur des points de convergence du trafic Web, ils mémorisent les documents transférés pour de futures requêtes. En effet un document déjà demandé une fois a plus de chances d'être redemandé qu'un document n'ayant jamais été accédé. Si un client demande un document qui a été mémorisé par le serveur mandataire, ce dernier le lui envoie immédiatement. Sinon, la requête est renvoyée vers le réseau, vers le serveur d'origine ou un serveur mandataire de niveau supérieur.

Les Proxy peuvent être transparents ou non. Un Proxy transparent s'utilise, comme son nom l'indique, comme s'il n'existait pas et les clients n'ont pas de configuration particulière à effectuer. Les Proxy non transparents, en revanche, obligent les utilisateurs à configurer leur navigateur pour qu'il utilise le Proxy.

Les Proxy actuels disposent d'algorithmes de décision avancés qui leur permettent de savoir quels documents doivent être conservés dans le cache et ceux qui doivent en être retirés. Pourtant, ces algorithmes ne sont adaptés qu'aux documents statiques, changeant peu souvent ou susceptibles d'être requis par de nombreux utilisateurs desservis par le même Proxy. En effet, un document modifié doit d'abord être transféré depuis le serveur d'origine et consomme donc de la bande passante. Un document devenant obsolète avant que le cache ait le temps de le resservir une seconde fois affiche même des performances de transfert moindres que s'il avait été directement récupéré depuis le serveur d'origine, du fait des traitements de recherche dans le cache et de mémorisation.

Et malheureusement, le contenu régulièrement visité par les internautes est justement celui qui a tendance à évoluer rapidement, et constitue une part croissante des sites Internet qui tirent parti des technologies qui permettent de rendre un site dynamique.

Ces contenus dynamiques constituent un nouveau problème pour les concepteurs de serveurs mandataires avec cache. Comment réduire l'utilisation de la bande passante pour les contenus dynamiques, de plus en plus populaires ?

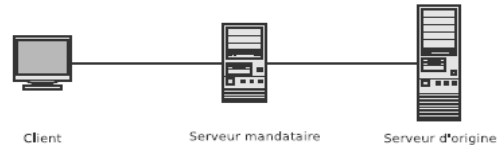


Figure 1. Réseau avec cache mandataire

2.2 Problématiques

La mise en cache des données consultées sur le réseau peut s'étudier sous divers angles, voici les principales orientations que les études prennent :

2.2.1 Économie de bande passante

La limitation de la bande passante utilisée pour le trafic Internet est une des principales préoccupations des propriétaires d'infrastructures. Cette problématique est valide à tous les niveaux du réseau, et encore plus pertinente dans le cadre d'organisations qui ne disposent que de connexions aux capacités limitées, ou qui louent des lignes spécifiques ; et peut tout autant s'appliquer dans le cadre de communications internes.

Pourtant, les caches ne sont pas toujours utilisés en priorité pour réduire l'utilisation de la bande passante. Par effet de bord les caches apportent ou peuvent apporter d'autres avantages, notamment des temps de réponse faibles. Cette étude portant sur l'optimisation des flux Web, cela sous-entend que l'on recherche une combinaison élevée de plusieurs facteurs et ne se limite pas à la limitation de l'utilisation maximale de l'infrastructure. De nombreux axes de recherche se focalisent sur la diminution du temps de réponse, ainsi que d'autres aspects alternatifs.

2.2.2 Temps de réponse

Plusieurs systèmes étudient la problématique des caches dans l'optique de l'optimisation du temps de réponse vis à vis de l'utilisateur. Cette approche s'oppose à la première car la principale technique utilisée pour diminuer le temps de chargement perçu par l'utilisateur consiste en un préchargement à l'avance des pages susceptibles d'être demandées par l'utilisateur dans le futur, ce qui induit une surconsommation de bande passante. De plus, ces systèmes ont souvent pour cible le navigateur du client, et se focalisent donc sur un usage mono-utilisateur où les priorités entre requêtes sont simples à déterminer, à savoir une priorité basse pour les données en préchargement et une priorité haute pour les requêtes effectivement formulées par l'utilisateur. Dans ce contexte mono-utilisateur, le concept d'optimisation de bande passante est aussi différent : il consiste à la saturer au possible pour maximiser son rendement pendant le temps de la connexion (notamment dans le cas des connexions RTC). Ce point de vue ne s'applique plus dans un environnement Multi-Utilisateurs (la notion de priorité ne peut plus être appliquée, car les utilisateurs

choisiraient systématiquement la priorité la plus élevée possible) où chacun partage une partie de la bande passante. Cependant, de nombreux travaux proposent des techniques pour anticiper avec une meilleure acuité les liens qui seront effectivement suivis par l'utilisateur, plutôt que de simplement précharger de nombreux liens. Certaines de ces techniques peuvent toutefois être réutilisées dans notre perspective.

2.2.3 Pertinence

Certains travaux proposent d'aider les utilisateurs à mieux trouver leur chemin à travers le dédale informationnel que constitue le réseau Internet. Ainsi, par effet de bord et non par objectif premier, la bande passante sera moins utilisée pour le transfert de documents pour lesquels l'utilisateur n'aura pas d'intérêt au final.

2.3 Objets non cachables

Il existe toujours des objets que l'on ne peut pas (ou doit pas) conserver dans un cache : ce sont les objets de taille trop importante (qui forceraient le cache à supprimer un grand nombre de documents plus petits), les objets générés dynamiquement en fonctions de paramètres spécifiques à l'utilisateur (par exemple des sites avec une identification comme un webmail), et en règle générale tous les sites qui utilisent des cookies pour conserver l'état d'une session (sites de shopping en ligne). Les connexions sécurisées par cryptage ne peuvent bien évidemment pas participer à la mise en cache.

3 De nouvelles solutions

Si les problèmes de bande passante et de temps de réponse ont été largement traités à ce jour, l'évolution du Web vers des sites dynamiques pouvant changer plus souvent que leur fréquence de consultation derrière un proxy, l'utilité du service de mise en cache redevient quasi nulle. Selon toute vraisemblance, on ne pourra pas «économiser» la mise à jour effective du document à partir du serveur d'origine au moment de la requête.

Face à ces faits, deux approches complémentaires sont à envisager : la limitation de la quantité de données à transférer et la détermination d'un moment plus propice pour effectuer ce transfert.

3.1 Mise à jour limitée

Pour diminuer l'importance des données à transférer lors d'un défaut de cache, quelques mécanismes ont été proposés :

- modifier le protocole http pour qu'il permette de transférer uniquement les différences entre deux documents (Network Working Group, 1999),
- utiliser des techniques de structuration du HTML pour séparer les documents en sous

documents plus petits plus faciles à mettre en cache,

Il est à noter que ces techniques nécessitent des modifications substantielles dans les protocoles et ne résolvent pas le problème de la bande passante nécessaire au transfert des documents qui changent souvent.

3.2 Mise à jour différée

Mettre à jour un cache lors de la pleine utilisation du réseau est inopportun. Cela allonge le temps nécessaire pour terminer la mise à jour et gêne les utilisateurs. On doit donc différer ces mises à jour, en les avançant dans le temps (et non en les retardant comme le suggère le terme).

3.2.1 Périodes creuses et pleines

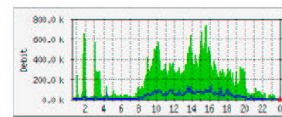


Figure 2. Utilisation journalière du réseau de l'UTBM

La plupart des réseaux se caractérisent par des périodes d'utilisation active et des périodes de repos (notamment pour le trafic HTTP). La figure 2 montre le trafic moyen à l'UTBM au cours d'une journée. Cette répartition de l'usage de la bande passante est assez classique et se retrouve dans de nombreux autres réseaux (Maltzahn & al., 1998). Pendant les périodes de pointe le trafic peut saturer les équipements, alors que les périodes creuses laissent les infrastructures inutilisées. La nature globalement irrégulière du trafic permet de diviser les périodes de creux en deux catégories :

- les creux réguliers, prévisibles, qui se situent la nuit et pendant la pause de midi,
- les creux fortuits, qui peuvent durer quelques secondes à plusieurs minutes, se produisent au hasard car les utilisateurs du réseau ne se concertent pas pour l'utilisation des ressources et les utilisent de manière prévisible globalement, mais très imprévisible si on cherche à augmenter la précision temporelle de cette utilisation.

3.2.1 Mise à profit des creux

Le dimensionnement correct d'un réseau tient compte non pas de la moyenne de son utilisation mais du maximum de son utilisation, sans quoi les périodes d'utilisation maximale seraient très pénibles pour les utilisateurs.

Il faut donc profiter de chacune de ces périodes de faible utilisation du réseau pour télécharger et mettre en cache à l'avance des contenus qui seront effectivement requis plus tard. Toute la difficulté réside dans la sélection des documents qui vont être préchargés.

Ayant maintenant la certitude qu'il existe des moments où la faible utilisation du réseau est propice à la mise à jour des caches, il faut maintenant pouvoir choisir les documents à effectivement mettre à jour, car le temps de faible utilisation n'est pas suffisant pour mettre à jour en intégralité le contenu du serveur mandataire. Pour cela on peut se baser sur les informations de modification provenant des serveurs d'origine des documents et les horaires de consultation effective des documents du cache.

3.3 Caches inverses

L'existence d'en-têtes spécifiques dans le protocole HTTP (Network Working Group, 1999) pour savoir si le cache peut servir un document ou s'il doit le télécharger depuis le serveur d'origine n'est pas toujours suffisante car le mécanisme demande tout de même des connexions montantes pour aller vérifier l'état du document. En effet une des opérations les plus coûteuses sur Internet est l'établissement d'une connexion. Même si l'étude des caches a aussi permis de mettre en place des mécanismes simples qui améliorent la performance (anticipation de la résolution DNS, de la connexion HTTP, «préchauffage» du serveur avec une requête vide et réutilisation des connexions TCP pour effectuer plusieurs requêtes HTTP), cette attitude d'attente active gaspille des ressources.

Le principe du cache inverse est de prévenir les serveurs mandataires que les documents dans leur cache ont changé. Plusieurs propositions concrètes de protocoles ont été faites, notamment le protocole WCIP (Li & al., 2001) qui permet aux serveurs mandataires de s'abonner à des canaux qui diffusent des messages d'invalidation ou de mise à jour. En fonction du protocole, la réception d'une notification d'invalidation peut soit informer le serveur mandataire qu'il devra télécharger une nouvelle version du document à la prochaine requête, mais cela peut être fait à tout moment. Dans le cas de diffusion de mises à jour, le cache n'aura pas besoin de récupérer de nouvelle version. Pourtant, ces deux scénarios restent potentiellement sous-optimaux, car ils peuvent récupérer le document à un moment de forte utilisation du réseau ou le transfèrent alors qu'il ne sera pas demandé.

3.3 Données de travail

La base de travail quasi exclusive des systèmes d'optimisation de cache sont les traces de l'activité du serveur (également appelés logs) des documents accédés depuis le plus longtemps possible. Sur ces traces seront effectuées diverses analyses statistiques de data mining destinées à extraire des règles ou des modèles récurrents, qui permettront, en fonction des caractéristiques des documents présents dans le cache, des pages visitées récemment et du moment de la journée et de l'état effectif du document original (obsolète ou non), de déclencher des mises à jour ou non.

4 Caches proactifs

Il existe de nombreux systèmes se rapprochant de la thématique des caches proactifs, dont quelques exemples sont présentés ci-dessous. Le plus souvent, ce sont des agents qui trouvent leur place sur le poste de l'utilisateur final (sous forme de plugins ou de programmes fonctionnant en parallèle du navigateur). Quelques-unes des fonctionnalités de ces agents sont parfois implémentés dans des serveurs de cache spécialisés.

Globalement, il y a deux principales classes de données qu'il est possible de mettre en cache à l'avance :

- Les données issues de liens connexes. Elles sont susceptibles d'intéresser le visiteur au cours de sa session de navigation ou d'une session ultérieure. Cette classe sera approfondie dans la section 6.
- Les données revalidées. Si on parvient à déterminer qu'un utilisateur consulte régulièrement un site ou une séquence de sites, il est possible de précharger les pages avant leur consultation, mais dans un creux d'utilisation du réseau et le plus tard possible (pour assurer que ce sera la dernière version qui sera fournie). Cette classe sera approfondie dans la section 5.

4.1 Agents d'accélération

Les plus simples agents d'«accélération» de navigation se contentent d'analyser les pages visitées par l'utilisateur et de suivre à l'avance les liens qui s'y trouvent. Des systèmes de cache, comme Wcol (Chinen & al., 1996) ou wwwoffle (Bishop), un cache pour petits réseaux, voire même mono-utilisateur fournissent cette fonctionnalité.

Il faut garder à l'esprit que nombre de liens présents sur la plupart des pages présuppose une quantité de pages à télécharger croissant de façon exponentielle, si aucune sélection n'est faite sur les liens à suivre. Des limitations triviales de l'espace des pages à télécharger, comme le domaine d'origine, ne pourront jamais être efficaces dans tous les cas. Il faut contrôler avec minutie la priorité des pages en attente de téléchargement : l'algorithme doit bien sûr effectuer une exploration en largeur d'abord et non en profondeur d'abord, et l'exploration des liens connexes sur les pages visitées plus récemment par l'utilisateur doit être prioritaire par rapport à ceux des pages plus anciennes.

Il est aussi à noter que la pratique de ce genre de balayage sauvage de tout un pan du web n'est généralement pas très apprécié des administrateurs, et que certains vont même jusqu'à installer des pièges qui détectent les robots et les bannissent du site.

4.2 Aides à la recherche et à la consultation

D'autres agents, plus «intelligents», sont capables d'explorer le web parallèlement à l'utilisateur pour lui

suggérer des pages susceptibles de l'intéresser. Ces agents utilisent des classificateurs sémantiques pour déterminer le sujet d'une page, et par exemple mettre dynamiquement en valeur les contenus intéressants ou ayant changé depuis la dernière visite.

4.3 Identification de l'utilisateur

On peut considérer tout le réseau derrière un proxy comme étant un utilisateur unique, très actif. Néanmoins quelques études montrent que l'on peut discerner les modèles utilisateur spécifiques et le modèle de l'utilisateur moyen, notamment en ce qui concerne les intérêts personnels de l'utilisateur, qui peuvent être utilisés dans les systèmes de cache qui explorent le web à l'avance.

S'il est possible de lier chaque utilisateur à une adresse IP (son poste de travail), alors il est possible facilement de différencier chaque utilisateur. En revanche, sur un réseau où les utilisateurs sont itinérants, il faut ajouter un mécanisme qui met à jour une table donnant les paires (utilisateur, IP) au moment où l'utilisateur s'authentifie.

Dans le cas d'une utilisation mutualisée d'un serveur, ou généralement si plusieurs utilisateurs sont simultanément connectés sur une machine, le problème devient bien plus complexe. Dans ce cas, soit on utilise un proxy non transparent, soit il faut utiliser des techniques d'identification par utilisateur (Leblond & al).

5 Prévisions à long terme et Habitudes des utilisateurs

Des études visant à améliorer le confort de l'utilisateur dans son utilisation du web ont proposé l'utilisation d'algorithmes issus de l'analyse statistique et du domaine de l'identification de modèles utilisateur. Il s'agit, parmi la masse d'informations que représente la totalité des traces d'accès du serveur, de dégager des règles explicites qui permettent de prédire les actions de l'utilisateur.

5.1 Modèle en découpage horaire

Une technique suggérée par le système Web Montage (Anderson & al., 2002) est le découpage des journées en «zones» horaires. Chaque zone est à mettre en relation avec la même zone les jours précédents, et on calcule ensuite le taux de popularité t_i de chaque URL trouvée dans les traces :

$$t_i = \sum_{j=1}^k \frac{a_{ij}}{k} \quad (1)$$

où k est le nombre de zones horaires prises en considération, et a_{ij} vaut 1 si l'URL i a été visitée dans la tranche horaire j . On pourrait aussi compter le nombre de fois qu'une page a été accédée dans une tranche horaire mais une visite massive à un moment donné fausserait les calculs. Par ailleurs, il y a beaucoup de manières pour améliorer la fiabilité de cette fonction, à commencer par y ajouter

un coefficient qui prend en compte l'ancienneté des traces :

$$t_i = \sum_{j=1}^k \frac{f(j) \times a_{ij}}{k} \quad (2)$$

Une simple fonction linéaire ou logarithmique peut donner des résultats satisfaisants ; il faut pourtant se souvenir que les traces anciennes ne doivent pas trop influencer le système mais qu'une irrégularité fortuite ne doit pas non plus trop le perturber.

Une limitation évidente de la méthode apparaît si on choisit des tranches horaires adjacentes : une requête régulière se produisant aux limites des tranches ne sera pas détectée car un utilisateur humain présentera forcément des irrégularités dans ses consultations. La solution est de choisir des tranches horaires qui se chevauchent, par exemple de 10h à 11h et de 10h30 à 11h30.

Un problème important reste est la difficulté de choix des tranches horaires et leur impact certain sur les résultats : des zones horaires trop grandes donneraient très peu d'informations au système, mais une granularité trop fine trouverait peu de résultats positifs si les visites ne sont pas suffisamment régulières. D'autre part l'utilisation de tranches horaires superposées nécessite plus de temps de traitement, et il faut gérer les visites régulières qui sont détectées dans plusieurs tranches horaires.

Avec une capacité de calcul suffisamment grande, il est possible de tester plusieurs niveaux de superposition de zones horaires, ce qui permet de détecter assez finement les horaires des visites régulières.

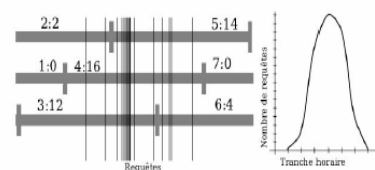


Figure 3. Algorithme de détection de visites récurrentes

Si le nombre de requêtes dépasse un certain seuil, on marque cette URL comme étant préchargeable.

5.2 Modèle statistique générique

Il est également d'utiliser des systèmes d'apprentissage génériques pour estimer quels contenus précharger avant le prochain pic d'utilisation [Erreur ! Source du renvoi introuvable.]. Cela ressemble un peu à une approche par système expert.

5.3 Le système d'apprentissage

Le système d'apprentissage se base sur un ensemble de variables et d'une décision (pour un système de cache : «télécharger» ou «ne pas télécharger»). Ensuite, à partir d'un ensemble d'exemples, comme des URLs qui ont été demandées par les utilisateurs

qui permettent de donner une valeur aux variables de décision, le système est capable d'exprimer les règles génériques qui régissent les exemples.

Un point intéressant de ce système d'apprentissage est qu'il n'exige pas que l'ensemble des exemples soit cohérent, c'est à dire que tous les exemples soient en accord. Dans ce cas, il se peut que quelques exemples contredisent les règles, mais comme le but du système est de trouver des règles génériques, l'impact sur les résultats n'est pas trop important.

Un exemple abstrait sur le jeu de golf est donné en Figure 4.

Décision et variables :

«Jouer», «Ne pas jouer».
prévision météo: «Ensoleillé», «Couvert», «Pluie».
température: continu.
humidité: continu.
venteux: «Vrai», «Faux».

Exemples :

«Ensoleillé»,85,85,Faux,«Ne pas jouer».
«Ensoleillé»,80,90,Vrai,«Ne pas jouer».
«Couvert»,83,78,«Faux»,«Jouer».
«Pluie»,70,96,«Faux»,«Jouer».
«Pluie»,68,80,«Faux»,«Jouer».
«Pluie»,65,70,«Vrai»,«Ne pas jouer».
«Couvert»,64,65,«Vrai»,«Jouer».
«Ensoleillé»,72,95,«Faux»,«Ne pas jouer».
«Ensoleillé»,69,70,«Faux»,«Jouer».
«Pluie»,75,80,«Faux»,«Jouer».
«Ensoleillé»,75,70,«Vrai»,«Jouer».
«Couvert»,72,90,«Vrai»,«Jouer».
«Couvert»,81,75,«Faux»,«Jouer».
«Pluie»,71,80,«Vrai»,«Ne pas jouer».

Règles :

«Ne pas jouer» :- venteux=«Vrai»,
prévision météo=«Pluie» (2/0).
«Ne pas jouer» :- humidité>=85,
prévision météo=«Ensoleillé» (3/0).
default : «Jouer» (9/0).

Figure 4. Exemple de sortie du système d'apprentissage. Les variables peuvent prendre des valeurs discrètes, ou continues (mot-clé 'continu'). Aucun exemple n'est ici en contradiction avec les règles extraites. Les indices entre parenthèses indiquent le nombre d'exemples en accord et en désaccord avec chaque règle

5.4 Précharger ou ne pas précharger

Les données permettant de savoir s'il faut ou non précharger se divisent en deux groupes :

- les entrées du cache qui ont été demandées dans la dernière période de forte utilisation du réseau et qui ont pu correctement être mises en cache sont des candidats au préchargement,

- les entrées qui ne satisfont pas les critères de mise en cache (objets non cachables, objets introuvables, objets modifiés entre le moment où ils ont déjà été préchargés une fois mais modifiés avant d'être demandés par un utilisateur, ou objets qui n'ont plus été demandés lors de la dernière période d'utilisation du réseau) permettent de connaître les documents qu'il ne faut pas télécharger.

5.5 Variables de décision

En utilisant comme variables de décision l'âge du document dans le cache, son type MIME, sa taille, le nom du serveur d'origine, son top level domain (.com, .net, .org ...), on obtient un ensemble qui peut être soumis au logiciel d'apprentissage. L'étude d'origine n'incluait pas l'heure d'accès aux documents dans les variables de décision, il n'est donc pas possible de savoir quel serait l'impact de cette donnée supplémentaire.

5.6 Résultats

Les essais réels de cette méthode utilisant l'apprentissage informatique ont montré qu'elle permettait de réduire substantiellement le niveau de bande passante nécessaire pour satisfaire les périodes de forte utilisation du réseau, et était capable de précharger jusqu'à 40% des documents demandés dans les périodes de forte utilisation, sans utiliser démesurément la bande passante lors des périodes de faible utilisation.

6 Prédiction à court terme

Dans leurs sessions de visite du web, les utilisateurs consultent souvent une série de sites «préférés». Si l'on connaît cette séquence, il est possible de précharger des documents en avance, ainsi l'accès semblera quasi instantané pour le client. De la même manière, un client est susceptible de suivre des liens vers des pages qui correspondent à ses centres d'intérêt. Une classification sémantique des pages permet alors de savoir s'il faut précharger les liens ou si l'utilisateur ne sera pas intéressé, et donc ne chargera pas la page.

6.1 Prédiction des séquences

On trouve de nombreuses études (Zhang & al., 2003) (Bouras & al., 2003) qui se concentrent sur la prédiction des séquences. Une des premières se concentrait non pas sur les URL mais les commandes unix (Davison & al., 1998), et avait pour but d'obtenir un algorithme générique de prédiction du prochain élément dans une séquence.

L'utilité d'un tel algorithme est qu'il permet de sélectionner relativement finement, parmi les nombreuses possibilités de liens qui s'offrent à l'utilisateur, ceux qu'il est le plus susceptible de suivre en fonction de son historique.

6.1.1 C4.5

L'algorithme de référence dans le domaine de la prédiction du prochain élément est C4.5. Néanmoins pour s'initialiser, cet algorithme a besoin de lire l'historique complet, avant de pouvoir faire des prédictions. Il ne fait pas de différence entre les éléments les plus anciens et les éléments les plus récents, et n'est pas capable de se mettre à jour en fonction des nouveaux événements, ni en fonction de l'exactitude ou l'inexactitude de ses prédictions.

6.1.2 IPAM

L'algorithme IPAM (Incremental Probabilistic Action Modeling) a été proposé pour fournir une version plus performante et plus flexible que C4.5 : cet algorithme est capable de se mettre à jour dynamiquement et affiche des performances supérieures à l'algorithme de référence.

6.1.3 PPM

Les algorithmes de prédiction par correspondance partielle (Prediction by Partial Match (Palpanas, 1998)) sont inspirés des algorithmes de compression de données utilisant des modèles de Markov pour prévoir statistiquement les valeurs dans un fichier. Dans le cadre du préchargement de données du web, l'algorithme utilise les chaînes de Markov pour prédire quelles sont les pages les plus susceptibles d'être demandées par les utilisateurs.

6.1.4 Utilisations

Ces techniques de prédiction interactives ont été utilisées dans plusieurs cas :

- un plugin pour le navigateur Mozilla qui utilise la bande passante inutilisée par l'internaute pour précharger les pages qu'il est susceptible de visiter [**Erreur ! Source du renvoi introuvable.**]. Ce plugin est effectivement un outil utilisant un véritable algorithme de prédiction et non un des nombreux logiciels qui se veulent des Web Accelerators et qui se contentent de télécharger sauvagement tous les liens présents sur une page ;
- Une étude sur le réseau GRNET en grèce a utilisé l'algorithme PPM pour implémenter un cache avec préchargement [**Erreur ! Source du renvoi introuvable.**], et certains des résultats expérimentaux de cette étude sont très prometteurs : par exemple, pour une augmentation de trafic inférieure à 2%, les défauts de cache se limitent à 25%.

6.2 Analyse des centres d'intérêt des utilisateurs

Dans quelques outils d'aide à l'organisation des signets (Li & al., 1999) et de génération de portails liés aux centres d'intérêt des utilisateurs [**Erreur ! Source du renvoi introuvable.**], des classifieurs sémantiques sont utilisés pour déterminer si une page est susceptible d'intéresser ou non l'internaute.

Ces classifieurs peuvent être utilisés pour prédire de manière plus précise si un utilisateur va suivre des

liens sur une page, ce qui est plus probable si la page l'intéresse.

7 Multimédia

Les téléchargements de gros fichiers, notamment vidéo, représentent aussi une part importante du trafic. Malheureusement, rien dans le protocole HTTP ne permet de mutualiser le transfert d'un fichier si plusieurs clients le demandent en même temps. Même si le fichier est mis en cache, ce qui est loin d'être certain s'il est très gros ou s'il n'est pas cachable, comme le streaming vidéo qui n'utilise pas le protocole HTTP, le dernier tronçon du réseau serait sur-utilisé car les mêmes données transitent deux fois.

Pour résoudre ce problème de sur-utilisation du réseau on peut, comme dans l'architecture Cyclone (Rost & al., 2001), utiliser un buffer circulaire qui indique quelle partie du fichier est en cours de téléchargement. Ainsi, un client qui demande le fichier alors que le serveur est déjà en train de l'envoyer devra commencer le téléchargement au milieu du flux. Le début du fichier sera envoyé plus tard, quand le premier client aura fini son téléchargement. Ainsi, le serveur ne doit ouvrir qu'un unique flux montant pour servir des milliers de clients. La distribution finale utilise la technique du multicast, qui permet d'envoyer un paquet à plusieurs destinataires

Conclusion

Cette étude a montré qu'en utilisant des algorithmes avancés, il est possible d'améliorer substantiellement la performance des systèmes de mise en cache, et de diminuer le niveau de bande passante maximale nécessaire dans un réseau.

De plus, il est possible d'améliorer le confort des utilisateurs qui ne bénéficient pas des systèmes de cache, par exemple ceux qui visitent régulièrement certains sites mais sont les seuls à les visiter.

En combinant ces diverses techniques, il est possible d'obtenir un système donnant une bonne approximation d'un système idéal qui aurait une parfaite connaissance du futur.

Références

- (Abrams & al., 1995) Marc Abrams, Charles R. Standridge, Ghaleb Abdulla, Stephen Williams et Edward A. Fox. Caching proxies: Limitations and potentials. Technical report, Virginia Tech, October 1995.
- (Anderson & al., 2002) Corin R. Anderson et Eric Horvitz. Web montage: A dynamic personalized start page, Mai 2002. <http://www2002.org/CDROM/refereed/468/>
- (Bishop) Andrew M. Bishop. World wide web offline. <http://www.gedanken.demon.co.uk/wwwoffline/>
- (Bouras & al., 2003) Christos Bouras, Agisilaos Konidaris, et Dionysos Kostoulas. Efficient reduction of web latency through predictive prefetching on a wan. Technical report, Research Academic Computer

- Technology Institute and Computer Engineering and Informatics Department, Patras, Greece, 2003.
- (Chinen & al., 1996) Ken-ichi Chinen, Suguru Yamaguchi. An interactive prefetching proxy server for improvement of www latency. Technical report, nara Institute of Science and Technology, Japon, 1996.
- (Davison & al., 1998) Brian D. Davison et Haym Hirsh. Predicting sequences of user actions. Technical report, University of New Jersey, 1998.
- (Leblond & al) Eric Leblond, Vincent Deffontaines. NuFW, Now User Filtering Works. <http://www.nufw.org/>.
- (Li & al., 2001) D. Li, P. Cao et M. Dahlin. Wcip : Web cache invalidation protocol. IETF Internet DRAFT, Mars 2001.
- (Li & al., 1999) Wen-Syan Li et Al. powerbookmarks: A system for personalizable web information organization, sharing and management, 1999.
- (Maltzahn & al., 1998) Carlos Maltzahn, Kathy J. Richardson, Dirk Grunwald, et James H. Martin. On bandwidth smoothing. Technical report, Compaq Computer Corporation, Network Systems Laboratory, 1998.
- (Mogul & al., 1997) Jeffrey C. Mogul, Fred Douglass, Anja Feldmann et Balachander Krishnamurthy. Potential benefits of delta encoding and data compression for http. Technical report, Digital Western Research Laboratory, 1997.
- (Network Working Group, 1999) Network Working Group. Hypertext transfer protocol http/1.1. Technical report, IETF/W3C, 1999.
- (Palpanas, 1998) Themistokolis Palpanas. Web prefetching using Partial Match Prediction. PhD thesis, Graduate Department of Computer Science, University of Toronto, 1998.
- (Rost & al., 2001) Stanislav Rost, John Byers, Azer Bestavros. The cyclone server architecture : Streamlining delivery of popular content. Technical report, Dept. of Computer Science, Boston University, Massachusetts, 2001.
- (Zhang & al., 2003) Whei Zhang, David B. Lewanda, Christopher D. Janneck, Brian D. Davison. Personalized web prefetching in mozilla. Technical report, Department of Computer Science and Engineering, Leigh University, 2003.