L'anatomie d'un moteur de recherche web hypertextuel à grande échelle

Sergey Brin, Lawrence Page

Résumé : Dans cet article, nous présentons Google, un prototype de moteur de recherche à grande échelle qui utilise largement la structure présente dans l'hypertexte. Google est conçu pour explorer et indexer le Web efficacement et produire des résultats de recherche beaucoup plus satisfaisants que les systèmes existants. Le prototype, doté d'une base de données de texte intégral et d'hyperliens d'au moins 24 millions de pages, est disponible à l'adresse http://google.stanford.edu/.

Concevoir un moteur de recherche est une tâche difficile. Les moteurs de recherche indexent des dizaines, voire des centaines de millions de pages Web impliquant un nombre comparable de termes distincts. Ils répondent à des dizaines de millions de requêtes chaque jour. Malgré l'importance des moteurs de recherche à grande échelle sur le Web, très peu de recherches universitaires ont été menées à leur sujet. De plus, en raison des progrès rapides de la technologie et de la prolifération du Web, la création d'un moteur de recherche Web aujourd'hui est très différente de celle d'il y a trois ans. Cet article fournit une description approfondie de notre moteur de recherche Web à grande échelle, la première description publique détaillée de ce type que nous connaissions à ce jour.

Outre les problèmes liés à l'adaptation des techniques de recherche traditionnelles à des données de cette ampleur, de nouveaux défis techniques sont liés à l'utilisation des informations supplémentaires présentes dans l'hypertexte pour produire de meilleurs résultats de recherche. Cet article aborde la question de savoir comment construire un système pratique à grande échelle capable d'exploiter les informations supplémentaires présentes dans l'hypertexte. Nous examinons également le problème de la gestion efficace des collections hypertextes non contrôlées où chacun peut publier ce qu'il veut.

1. Introduction

Le Web crée de nouveaux défis pour la recherche d'informations. La quantité d'informations sur le Web augmente rapidement, tout comme le nombre de nouveaux utilisateurs inexpérimentés dans l'art de la recherche sur le Web. Il est probable que les internautes surfent sur le Web grâce à son graphe de liens, en commençant souvent par des index de haute qualité, gérés par des personnes, comme Yahoo![1], ou par des moteurs de recherche. Ces listes, gérées par des personnes, couvrent efficacement les sujets populaires, mais sont subjectives, coûteuses à créer et à maintenir, lentes à améliorer et elles ne peuvent couvrir tous les sujets ésotériques. Les moteurs de recherche automatisés, qui s'appuient sur la correspondance de mots clés, renvoient généralement trop de résultats de mauvaise qualité. Pire encore, certains annonceurs tentent d'attirer l'attention en utilisant des méthodes visant à tromper les moteurs de recherche automatisés. Nous avons développé un moteur de recherche à grande échelle qui résout de nombreux problèmes des systèmes existants. En particulier, il exploite la structure supplémentaire de l'hypertexte pour fournir des résultats de recherche de bien meilleure qualité. Nous avons choisi pour dénommer notre système le mot Google, car

Référence : Computer Networks and ISDN Systems (Réseaux informatiques et systèmes) 30 (1998) 107-117.
Département d'informatique. Université de Stanford, Stanford, CA 94305, États-Unis.
Il existe deux versions de cet article : une version complète plus longue et une version à imprimer plus courte. La version complète est disponible sur le Web et sur le CD-ROM de la conférence.
Courriel : sergey, page@cs.stanford.edu
Transcription (et correction de la traduction automatique par Google traduction) :
Denise Vella-Chemla, juillet 2025.

[1] http://www.yahoo.com/.

il s'agit d'une orthographe courante du mot "*gogol*", qui désigne le nombre 10^{100}, et que ce mot correspond parfaitement à notre objectif de développer des moteurs de recherche à très grande échelle.

1.1. Moteurs de recherche pour le Web, l'expansion des années 1994-2000

La technologie des moteurs de recherche a dû évoluer considérablement pour suivre la croissance du Web. En 1994, l'un des premiers moteurs de recherche, le World Wide Web Worm (WWWW) [6], indexait 110 000 pages et documents accessibles sur le Web.

Dès novembre 1997, les principaux moteurs de recherche affirmaient indexer entre 2 millions (WebCrawler) et 100 millions de documents Web (d'après Search Engine Watch[2]). On peut prévoir que d'ici l'an 2000, un index complet du Web contiendra plus d'un milliard de documents[3].

Parallèlement, le nombre de requêtes traitées par les moteurs de recherche a lui aussi connu une croissance fulgurante. En mars et avril 1994, le World Wide Web Worm recevait en moyenne environ 1 500 requêtes par jour. En novembre 1997, Altavista affirmait traiter environ 20 millions de requêtes par jour. Avec l'augmentation du nombre d'utilisateurs sur le Web et l'automatisation des systèmes d'interrogation des moteurs de recherche, il est probable que les principaux moteurs de recherche traiteront des centaines de millions de requêtes par jour d'ici l'an 2000. L'objectif de notre système est de résoudre de nombreux problèmes, tant en termes de qualité que d'évolutivité, posés par l'extension de la technologie des moteurs de recherche à un niveau aussi extraordinaire.

1.2. Google : s'adapter au Web

Créer un moteur de recherche évolutif, même pour le Web actuel, présente de nombreux défis. Une technologie d'exploration rapide est nécessaire pour collecter les documents Web et les maintenir à jour. L'espace de stockage doit être optimisé pour stocker les index et, éventuellement, les documents eux-mêmes. Le système d'indexation doit traiter efficacement des centaines de gigaoctets de données. Les requêtes doivent être traitées rapidement, à une cadence de plusieurs centaines, voire de milliers, par seconde.

Ces tâches deviennent de plus en plus difficiles à mesure que le Web se développe. Cependant, les performances et le coût du matériel se sont considérablement améliorés, compensant partiellement cette difficulté. Il existe toutefois plusieurs exceptions notables à ces progrès, telles que le temps de recherche sur le disque et la robustesse du système d'exploitation. Lors de la conception de Google, nous avons pris en compte à la fois le taux de croissance du Web et les évolutions technologiques. Google est conçu pour s'adapter facilement à des ensembles de données extrêmement volumineux. Il utilise efficacement l'espace de stockage pour stocker l'index. Ses structures de données sont optimisées pour un accès rapide et efficace (voir section 4.2). De plus, nous prévoyons que le coût d'indexation et de stockage de texte ou de l'HTML diminuera éventuellement par rapport à la quantité disponible (voir l'annexe B de la version complète). Cela se traduira par des propriétés

[2] http://www.searchenginewatch.com.

[3] *Note de la traductrice : (trouvé sur un blog, fiabilité ?) 20 milliards de sites web sont crawlés et indexés par Google quotidiennement.*

d'évolutivité favorables pour les systèmes centralisés comme Google.

1.3. Objectifs de conception

1.3.1. Amélioration de la qualité de la recherche

Notre objectif principal est d'améliorer la qualité des moteurs de recherche Web. En 1994, certains pensaient qu'un index de recherche complet permettrait de trouver n'importe quoi facilement. Selon *Best of the Web 1994 - Navigators* [4], "Le meilleur service de navigation devrait permettre de trouver facilement presque tout sur le Web (une fois toutes les données saisies)". Cependant, le Web de 1997 est bien différent. Quiconque a utilisé un moteur de recherche récemment peut aisément témoigner du fait que l'exhaustivité de l'index n'est pas le seul facteur de qualité des résultats. Les "résultats indésirables" effacent souvent les résultats intéressants. De fait, en novembre 1997, un seul des quatre principaux moteurs de recherche commerciaux se retrouve lui-même (i.e. il renvoie sa propre page de recherche lorsque son nom apparaît dans les dix premiers résultats). L'une des principales causes de ce problème est que le nombre de documents dans les index a considérablement augmenté, contrairement à la capacité de l'utilisateur à les consulter. Les utilisateurs souhaitent toujours ne consulter que les premières dizaines de résultats. C'est pourquoi, à mesure que le nombre de données disponibles augmente, nous avons besoin d'outils très précis (nombre de documents pertinents renvoyés, par exemple dans les dix premiers résultats). En effet, nous souhaitons que notre notion de "pertinent" n'inclue que les meilleurs documents, car il peut y avoir des dizaines de milliers de documents légèrement pertinents. Cette très grande précision est importante, même au détriment de la réponse globale elle-même : la réponse globale est le nombre *total* de documents pertinents que le système est capable de renvoyer. Il existe récemment un certain optimisme quant à l'utilisation de davantage d'informations hypertextuelles pouvant contribuer à améliorer la recherche et d'autres applications [4.9.12.3]. En particulier, la structure des liens [7] et le texte des liens fournissent de nombreuses informations pour juger de la pertinence et filtrer la qualité. Google utilise à la fois la structure des liens et le texte d'ancrage (voir sections 2.1 et 2.2).

1.3.2. Recherche universitaire sur les moteurs de recherche

Outre sa croissance phénoménale, le Web est également devenu de plus en plus commercial au fil du temps. En 1993, 1,5 % des serveurs Web étaient sur des domaines dont les noms étaient suffixés par ".com". Ce chiffre est passé à plus de 60 % en 1997. Parallèlement, les moteurs de recherche ont migré du domaine universitaire vers le domaine commercial. Jusqu'à présent, la plupart des développements de moteurs de recherche ont été réalisés par des entreprises qui publiaient peu de détails techniques. De ce fait, la technologie des moteurs de recherche reste largement un art obscur et elle est orientée vers la publicité (voir l'annexe A de la version complète de notre article). Avec Google, nous avons pour objectif clair de promouvoir le développement et la compréhension dans le domaine universitaire.

Un autre objectif de conception important était de créer des systèmes qu'un nombre raisonnable de personnes puissent réellement utiliser. L'utilisation était importante pour nous, car nous pensons que certaines des recherches les plus intéressantes impliqueront l'exploitation de la vaste quantité de

[4] Voir http://botw.org/1994/awards/navigators.html.

données d'utilisation disponibles sur les systèmes Web modernes. Par exemple, plusieurs dizaines de millions de recherches sont effectuées chaque jour. Cependant, il est très difficile d'obtenir ces données, principalement parce qu'elles sont considérées comme ayant une valeur commerciale.

Notre objectif final était de construire une architecture capable de prendre en charge de nouvelles activités de recherche sur des données Web à grande échelle. Pour soutenir de nouvelles utilisations de la recherche, Google stocke tous les documents réels qu'il explore sous forme compressée. L'un de nos principaux objectifs lors de la conception de Google était de mettre en place un environnement où d'autres chercheurs peuvent intervenir rapidement, traiter de grandes quantités de données Web et produire des résultats intéressants qui auraient été très difficiles à obtenir autrement. Depuis que le système est opérationnel, plusieurs articles ont déjà été publiés utilisant des bases de données.

Un autre objectif de conception important était de créer des systèmes qu'un nombre raisonnable de personnes puissent réellement utiliser. L'utilisation était importante pour nous, car nous pensons que certaines des recherches les plus intéressantes impliqueront l'exploitation de la vaste quantité de données d'utilisation disponibles sur les systèmes Web modernes. Par exemple, plusieurs dizaines de millions de recherches sont effectuées chaque jour. Cependant, il est très difficile d'obtenir ces données, principalement parce qu'elles sont considérées comme ayant une valeur commerciale

Notre objectif final était de construire une architecture capable de prendre en charge de nouvelles activités de recherche sur des données Web à grande échelle. Pour soutenir de nouvelles utilisations de la recherche, Google stocke tous les documents réels qu'il explore sous forme compressée. L'un de nos principaux objectifs lors de la conception de Google était de mettre en place un environnement dans lequel d'autres chercheurs puissent intervenir rapidement, traiter de grandes quantités de données Web et produire des résultats intéressants qui auraient été très difficiles à obtenir autrement. Depuis que le système est opérationnel, plusieurs articles ont déjà été publiés utilisant des bases de données générées par Google, et bien d'autres sont en cours. Un autre de nos objectifs est de mettre en place un environnement de type Spacelab où les chercheurs, voire les étudiants, peuvent proposer et réaliser des expériences intéressantes sur nos données Web à grande échelle.

2. Fonctionnalités du système

Le moteur de recherche Google possède deux fonctionnalités importantes qui l'aident à produire des résultats de haute précision. Premièrement, il utilise la structure des liens du Web pour effectuer un classement de qualité de toutes les pages Web. Ce classement est appelé PageRank et il est décrit en détail dans [7]. Deuxièmement, Google utilise les liens pour améliorer les résultats de recherche.

2.1. PageRank : mettre de l'ordre sur le Web

Le graphe des citations (liens) du Web est une ressource importante qui est largement restée inutilisée dans les moteurs de recherche Web existants. Nous avons créé des cartes contenant jusqu'à 518 millions de ces hyperliens, soit un échantillon significatif du total. Ces cartes permettent un calcul rapide du "PageRank" d'une page Web, qui est une mesure objective du nombre de cita-

tions de cette page, qui correspond bien à l'idée subjective que les gens se font de l'importance d'une page. Grâce à cette correspondance, le PageRank est un excellent moyen de hiérarchiser les résultats des recherches par mots-clés sur le Web. Pour les sujets les plus populaires, une simple recherche par correspondance de texte limitée aux titres de pages Web fonctionne admirablement lorsque le PageRank donne la priorité aux résultats[5]. Pour le type de recherches en texte intégral dans le système principal de Google, le PageRank est également très utile.

2.1.1. Description du calcul du PageRank

La littérature sur les citations universitaires a été appliquée au Web, principalement en comptant les citations ou les liens pointant vers une page donnée. Cela donne une approximation de l'importance ou de la qualité d'une page. Le PageRank étend cette idée en ne comptant pas les liens de toutes les pages de manière égale et en normalisant par le nombre de liens sur une page. Le PageRank est défini comme suit :

> *Nous supposons que la page A est citée dans les pages $T_1, \ldots, T_n$ qui pointent vers elle (c'est-à-dire que les pages $T_1, \ldots, T_n$ contiennent des citations de la page A). Le paramètre d est un facteur d'amortissement, qui peut être défini entre 0 et 1. Nous définissons généralement d à $0,85$. On fournira plus de détails à propos de d dans la section suivante. $C(A)$ est également défini comme le nombre de liens sortant de la page A. Le PageRank d'une page A est donné comme suit :*
>
> $$PR(A) = (1-d) + d\left(\frac{PR(T_1)}{C(T_1)} + \ldots + \frac{Pr(T_n)}{C(T_n)}\right)$$
>
> *Notez que les PageRanks forment une distribution de probabilité sur les pages Web, de sorte que la somme des PageRanks de toutes les pages Web sera égale à un.*

Le PageRank ou $PR(A)$ peut être calculé à l'aide d'un algorithme itératif simple et correspond au vecteur propre principal de la matrice de liens normalisée du Web. De plus, un PageRank pour 26 millions de pages Web peut être calculé en quelques heures sur un poste de travail de taille moyenne. Il existe de nombreux autres détails qui dépassent le cadre de cet article.

2.1.2. Justification intuitive

Le PageRank peut être considéré comme un modèle de comportement utilisateur. Nous supposons qu'il existe un "internaute aléatoire" à qui l'on donne une page Web au hasard et qui continue à cliquer sur les liens, sans jamais appuyer sur "retour", mais l'internaute en question finit par se lasser et il recommence sur une autre page au hasard. La probabilité que l'internaute aléatoire visite une page est le PageRank de la page en question. Le facteur d'amortissement d représente la probabilité qu'à chaque page, l'internaute se lasse et aille sur une autre page aléatoire. Deux variantes importantes de l'algorithme consistent soit à n'appliquer le facteur d'amortissement d qu'à une seule page, soit à l'appliquer à un groupe de pages. Cela permet la personnalisation et rend quasiment impossible toute tentative délibérée de tromper le système pour obtenir un meilleur

[5] *Note de la traductrice : à l'époque de la sortie de l'article, une démonstration était accessible sur le site de l'université Stanford. Le lien fourni* `google.stanford.edu` *ne permet plus d'accéder à la page en question.*

classement. Il existe plusieurs autres extensions du PageRank, voir [7].

Une autre justification intuitive est qu'une page peut avoir un PageRank élevé si de nombreuses pages pointent vers elle, ou bien si certaines pages qui pointent vers elle ont un PageRank élevé. Intuitivement, les pages citées en de nombreux endroits sur le Web méritent d'être consultées. De même, les pages ne contenant qu'une seule citation, par exemple celle de la page d'accueil de Yahoo! [6], méritent généralement d'être consultées. Si une page n'était pas de bonne qualité ou comportait un lien rompu, il est fort probable que la page d'accueil de Yahoo n'y renvoie pas. Le PageRank gère ces deux cas et tous les cas intermédiaires en propageant récursivement les pondérations à travers la structure des liens du Web.

2.2. Texte d'ancrage

Le texte des liens est traité de manière particulière dans notre moteur de recherche. La plupart des moteurs de recherche associent le texte d'un lien à la page sur laquelle il se trouve. De plus, nous l'associons à la page vers laquelle le lien pointe. Cela présente plusieurs avantages. Premièrement, les ancres fournissent souvent des descriptions plus précises des pages Web que les pages elles-mêmes. Deuxièmement, des ancres peuvent exister pour des documents qui ne peuvent pas être indexés par un moteur de recherche textuel, tels que des images, des programmes et des bases de données. Cela permet de renvoyer des pages Web qui n'ont pas été réellement explorées. Notez que les pages qui n'ont pas été explorées peuvent poser des problèmes, car leur validité n'est jamais vérifiée avant d'être renvoyée à l'utilisateur. Dans ce cas, le moteur de recherche peut même renvoyer une page qui n'a jamais réellement existé, mais qui avait des hyperliens pointant vers elle. Cependant, il est possible de trier les résultats, de sorte que ce problème particulier se produit rarement.

Cette idée de propagation du texte d'ancrage à la page à laquelle il fait référence a été mise en œuvre dans le World Wide Web Worm [6], notamment parce qu'elle permet de rechercher des informations non textuelles et d'élargir la couverture de la recherche tout en téléchargeant moins de documents. Nous utilisons la propagation des ancres principalement parce que le texte d'ancrage peut contribuer à fournir des résultats de meilleure qualité. L'utilisation efficace du texte d'ancrage est techniquement difficile en raison des grandes quantités de données à traiter. Dans notre exploration actuelle de 24 millions de pages, nous avons indexé plus de 259 millions d'ancres.

3. Travaux connexes

La recherche sur le Web a une histoire courte et concise. Le World Wide Web Worm (WWWW) [6] a été l'un des premiers moteurs de recherche Web. Il a ensuite été suivi par plusieurs moteurs de recherche universitaires, dont beaucoup sont maintenant des sociétés cotées en bourse. Comparé à la croissance du Web et à l'importance des moteurs de recherche, il existe très peu de documents sur les moteurs de recherche récents [8]. Selon Michael Mauldin (directeur scientifique, Lycos Inc.) [5], "les différents services (y compris Lycos) gardent précieusement les détails de ces bases de données". Cependant, de nombreux travaux ont été consacrés aux fonctionnalités spécifiques des moteurs de recherche. Une fonctionnalité particulière souvent présentée permet d'obtenir des

[6] http://www.yahoo.com/.

résultats en post-traitant les résultats des moteurs de recherche commerciaux existants, une autre fonctionnalité consiste à produire des moteurs de recherche "individualisés" à petite échelle. Enfin, de nombreuses recherches ont été menées sur les systèmes de recherche d'information, en particulier sur les collections de données bien contrôlées [11].

Cependant, les travaux sur la recherche d'information ont principalement porté sur des collections de données assez petites et bien contrôlées, telles que l'ensemble de données de la Text Retrieval Conference [10]. Ce qui fonctionne bien sur TREC ne produit souvent pas de bons résultats sur le Web. Par exemple, le modèle d'espace vectoriel standard essaie de renvoyer le document qui se rapproche le plus de la requête, étant donné que la requête et le document sont des vecteurs définis par leur occurrence de mot. Sur le Web, cette stratégie renvoie souvent des documents très courts qui contiennent la requête plus quelques mots. Cela peut être le cas même pour un moteur de recherche très utilisé. Compte tenu des exemples que nous avons étudiés, nous pensons que le travail standard de recherche d'information doit être étendu pour traiter efficacement le Web.

Le Web est une vaste collection de documents hétérogènes totalement incontrôlés. Les documents varient considérablement en termes de langue, de format et de style. Il peut y avoir de nombreuses différences d'ordre de grandeur entre la taille, la qualité, la popularité et la fiabilité de deux documents. Tous ces éléments constituent des défis importants pour une recherche efficace sur le Web. Ils sont quelque peu influencés par la disponibilité de données auxiliaires telles que les hyperliens et le formatage, et Google essaie de tirer parti de ces deux éléments.

4. Anatomie du système

4.1. Présentation de l'architecture de Google

Dans cette section, nous donnerons un aperçu général du fonctionnement de l'ensemble du système, comme illustré à la figure 1. D'autres sections aborderont les applications et les structures de données non mentionnées dans cette section. La majeure partie de Google est implémentée en `C` ou `C++` pour plus d'efficacité et peut fonctionner sous Solaris ou Linux.

Chez Google, l'exploration du Web (le téléchargement de pages Web) est effectuée par plusieurs robots d'exploration distribués. Il existe un serveur d'URL qui envoie des listes d'URL à récupérer aux robots d'exploration. Les pages Web, i.e. les documents récupérés, sont ensuite envoyés au serveur de stockage. Le serveur de stockage compresse et stocke ensuite les pages Web dans un référentiel. Chaque page Web possède un numéro d'identification associé, appelé docID, qui est attribué chaque fois qu'une nouvelle URL est analysée à partir d'une page Web.

La fonction d'indexation est effectuée par l'indexeur et le trieur. L'indexeur exécute un certain nombre de fonctions. Il lit le référentiel, décompresse les documents et les analyse. Chaque document est converti en un ensemble d'occurrences de mots appelées hits. Les hits enregistrent le mot, la position dans le document, une approximation de la taille de la police et la casse. L'indexeur distribue ces hits dans un ensemble de "barils", créant ainsi un index direct partiellement trié. L'indexeur exécute une autre fonction importante. Il analyse tous les liens de chaque page Web et stocke des informations importantes à leur sujet dans un fichier d'ancres. Ce fichier contient

suffisamment d'informations pour déterminer d'où et vers où pointe chaque lien, ainsi que le texte du lien.

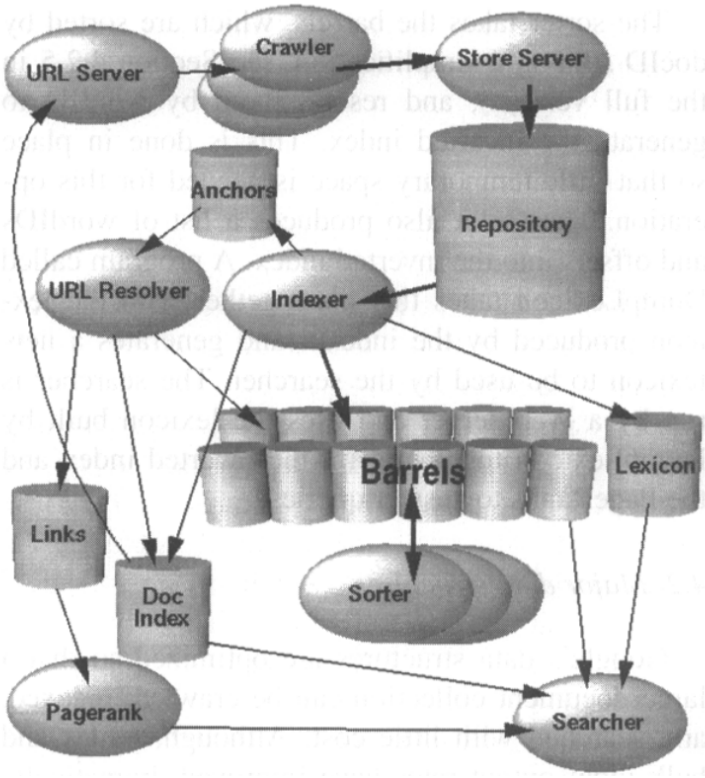


FIG. 1. Architecture de haut niveau de Google.

Le résolveur d'URL lit le fichier d'ancres et convertit les URL relatives en URL absolues, puis en docID. Il place le texte d'ancrage dans l'index vers l'avant (*forward index*), associé au docID vers lequel l'ancre pointe. Il génère également une base de données de liens qui sont des paires de docID. La base de données de liens est utilisée pour calculer les PageRanks de tous les documents.

Le trieur prend les "barils", triés par docID (il s'agit d'une simplification, voir la section 4.2.5 dans la version complète), et les trie par wordID pour générer l'index inversé. Cette opération est effectuée sur place afin de minimiser l'espace temporaire nécessaire. Le trieur produit également une liste de wordID et de décalages dans l'index inversé. Un programme appelé DumpLexicon prend cette liste ainsi que le lexique produit par l'indexeur et génère un nouveau lexique à utiliser par le chercheur. Le moteur de recherche est exécuté par un serveur Web et il utilise le lexique construit par DumpLexicon avec l'index inversé et les PageRanks pour répondre aux requêtes.

4.2. Principales structures de données

Les structures de données de Google sont optimisées pour qu'une grande collection de documents puisse être explorée, indexée et recherchée à faible coût. Bien que les processeurs et les débits d'entrée-sortie en masse se soient considérablement améliorés au fil des ans, une recherche sur disque nécessite encore environ 10 ms. Google est conçu pour éviter les recherches sur disque autant que possible, ce qui a eu une influence considérable sur la conception des structures de données.

La version complète de cet article contient une discussion détaillée de toutes les principales structures de données. Nous n'en donnons ici qu'un bref aperçu.

Presque toutes les données de Google sont stockées dans des Bigfiles, des fichiers virtuels que nous avons développés et qui peuvent s'étendre sur plusieurs systèmes de fichiers et prendre en charge la compression. Le référentiel HTML brut utilise environ la moitié de l'espace de stockage nécessaire. Il consiste en la concaténation du code HTML compressé de chaque page, précédé d'un petit entête. L'index des documents conserve des informations sur chaque document. Il s'agit d'un index ISAM (Index sequential access mode) à largeur fixe, classé par docID. Les informations stockées dans chaque entrée comprennent l'état actuel du document, un pointeur vers les référentiels, une somme de contrôle du document et diverses statistiques. Les informations de largeur variable telles que l'URL et le titre sont conservées dans un fichier séparé. Il existe également un index auxiliaire pour convertir les URL en docID. Le lexique a plusieurs formes différentes pour différentes opérations. Ce sont toutes des tables de hachage en mémoire avec des valeurs variables attachées à chaque mot.

Une liste de résultats (*hit list*) correspond à une liste d'occurrences d'un mot particulier dans un document particulier, y compris les informations de position, de police et de casse. Les listes de résultats (*hit lists*) occupent la majeure partie de l'espace utilisé dans les index directs et inversés. Il est donc important de les représenter le plus efficacement possible. Nous avons envisagé plusieurs alternatives pour coder la position, la police et la casse : un codage simple (un triplet d'entiers), un codage compact (une allocation de bits optimisée manuellement) et un codage de Huffman. Finalement, nous avons opté pour un codage compact optimisé manuellement, car il nécessite beaucoup moins d'espace que le codage simple et beaucoup moins de manipulations de bits que le codage de Huffman. Notre codage compact utilise deux octets pour chaque résultat. Les détails de ce codage sont disponibles dans la version complète de cet article. La longueur d'une liste de résultats est stockée avant les résultats eux-mêmes. Pour économiser de l'espace, la longueur de la liste de résultats est combinée avec l'identifiant du mot dans l'index direct et l'identifiant du document dans l'index inversé.

L'index vers l'avant est en fait déjà partiellement trié. Il est stocké dans un certain nombre de barils (nous en avons utilisé 64). Chaque baril contient une plage d'identifiants de mots. Si un document contient des mots qui appartiennent à un baril particulier, l'identifiant du document (*docID*) est enregistré dans le baril, suivi d'une liste d'identifiants de mots (*wordID*) avec des listes de résultats (*hit list*) correspondant à ces mots. Ce schéma nécessite un peu plus de stockage en raison des identifiants de document dupliqués, mais la différence est très faible pour un nombre raisonnable de compartiments et permet de gagner un temps considérable et de réduire la complexité du codage lors de la phase d'indexation finale effectuée par le trieur. L'index inversé est constitué des mêmes barils que l'index vers l'avant, sauf qu'ils ont été traités par le trieur. Pour chaque identifiant de mot valide, le lexique contient un pointeur vers le baril dans lequel se trouve l'identifiant de mot. Il pointe vers une liste d'identifiants de documents avec leurs listes de résultats correspondantes. Cette liste est appelée la *docList* d'un mot, elle représente toutes les occurrences de ce mot dans tous les documents.

Un problème important est de savoir dans quel ordre les *docID* (les identifiants de documents)

doivent apparaître dans la *docList* (la liste de documents). Une solution simple consiste à les stocker triés par *docID*. Cela permet une fusion rapide de différentes listes de documents pour les requêtes portant sur plusieurs mots. Une autre option consiste à les stocker triés par ordre d'occurrence du mot dans chaque document. Cela rend la réponse aux requêtes portant sur un seul mot triviale et rend probable que les réponses aux requêtes portant sur plusieurs mots soient plus proches de celles portant sur les mots du début de la liste. Cependant, la fusion est beaucoup plus difficile. De plus, cela rend le développement beaucoup plus difficile, car une modification de la fonction de classement nécessite une reconstruction de l'index. Nous avons choisi un compromis entre ces options, en conservant deux ensembles de barils inversés, un ensemble pour les listes de résultats, qui inclut les titres ou les ancres, et un autre ensemble pour toutes les listes de résultats. De cette façon, nous vérifions d'abord le premier ensemble de barils et s'il n'y a pas assez de correspondances entre les barils, on teste alors les barils plus grands.

4.3. Explorer le Web

Exécuter un robot d'exploration Web est une tâche difficile. On est confronté à des problèmes délicats de performance et de fiabilité et, plus important encore, à des problèmes sociaux. L'exploration est l'application la plus fragile car elle implique une interaction avec des centaines de milliers de serveurs Web qui ont des noms différents et qui échappent tous au contrôle du système.

Afin de gérer des centaines de millions de pages Web, Google dispose d'un système d'exploration distribué rapide. Un seul serveur d'URL fournit des listes d'URL à un certain nombre de robots d'exploration (nous en avons généralement environ 3 qui s'exécutent simultanément). Le serveur d'URL et les robots d'exploration sont tous deux implémentés en Python. Chaque robot d'exploration maintient environ 300 connexions ouvertes simultanément. Cela est nécessaire pour récupérer les pages Web à un rythme suffisamment rapide. À des vitesses de pointe, le système peut explorer plus de 100 pages Web par seconde en utilisant quatre robots d'exploration. La recherche DNS constitue une contrainte majeure en termes de performances. Chaque robot d'exploration conserve donc un cache DNS. Chacune des centaines de connexions peut se trouver dans différents états : recherche DNS, connexion à l'hôte, envoi de requête et réception de réponse. Ces facteurs font du robot d'exploration un composant complexe du système. Il utilise des E/S (entrées/sorties) asynchrones pour gérer les événements et un certain nombre de files d'attente pour faire passer les pages récupérées d'un état à l'autre.

Les plus d'un demi-million de serveurs que nous explorons sont gérés par des dizaines de milliers de webmasters. Par conséquent, explorer le Web implique d'interagir avec un bon nombre de personnes. Presque quotidiennement, nous recevons des courriels du type "Waouh, vous avez consulté de nombreuses pages de mon site Web. Vous en avez pensé quoi ?!". D'autres interactions impliquent des problèmes de droits d'auteur ou bien des bugs obscurs qui peuvent ne survenir que sur une page sur dix millions. Étant donné que les grands systèmes complexes tels que les robots d'exploration causent invariablement des problèmes, des ressources importantes doivent être consacrées à la lecture des courriels et à la résolution de ces problèmes au fur et à mesure qu'ils surviennent.

4.4. Recherche

L'objectif de la recherche est de fournir des résultats de recherche de qualité de manière efficace. La plupart des moteurs de recherche commerciaux semblent avoir fait de grands progrès en termes d'efficacité. Par conséquent, nous nous sommes davantage concentrés sur la qualité de la recherche dans nos recherches, même si nous pensons que nos solutions sont adaptables aux volumes commerciaux avec un peu plus d'efforts.

Google conserve beaucoup plus d'informations sur les documents Web que les moteurs de recherche classiques. Chaque liste de résultats comprend des informations sur la position, la police et la casse. De plus, nous prenons en compte les résultats du texte d'ancrage et le PageRank du document. Combiner toutes ces informations pour obtenir un classement est difficile. Nous avons conçu notre fonction de classement de manière à ce qu'aucun facteur ne puisse avoir trop d'influence. Pour chaque document correspondant, nous calculons le nombre de résultats de différents types à différents niveaux de proximité. Ces résultats calculatoires sont ensuite analysés dans une série de tables de recherche et sont finalement transformés en classement. Ce processus implique de nombreux paramètres réglables. Nous n'avons pas passé beaucoup de temps à régler le système ; nous avons plutôt développé un système de rétroaction qui nous aidera à ajuster ces paramètres à l'avenir.

5. Résultats et performances

La mesure la plus importante d'un moteur de recherche est la qualité de ses résultats. Bien qu'une évaluation complète par l'utilisateur dépasse le cadre de cet article, notre propre expérience avec Google a montré qu'il produit de meilleurs résultats que les principaux moteurs de recherche commerciaux pour la plupart des recherches. À titre d'exemple illustrant l'utilisation du PageRank, du texte d'ancrage et de la proximité, la figure 2 montre les résultats de Google pour une recherche sur "Bill Clinton". Ces résultats illustrent certaines des fonctionnalités de Google. Les résultats sont regroupés par serveur. Cela aide considérablement lors du tri des ensembles de résultats. Un certain nombre de résultats proviennent du domaine whitehouse.gov, ce que l'on peut raisonnablement attendre d'une telle recherche. Actuellement, la plupart des principaux moteurs de recherche commerciaux ne renvoient aucun résultat de `whitehouse.gov`, et les bons moteurs renvoient encore moins de résultats depuis cette source. Notez qu'il n'y a pas de titre pour le premier résultat. Google s'est plutôt appuyé sur le texte d'ancrage pour déterminer qu'il s'agissait d'une bonne réponse à la requête. De même, le cinquième résultat est une adresse e-mail qui, bien sûr, n'est pas accessible. Il est également le résultat du texte d'ancrage.

Tous les résultats sont des pages de qualité raisonnablement élevée et, lors de la dernière vérification, aucun n'était un lien rompu.

Cela s'explique en grande partie par leur PageRank élevé. Les PageRanks sont les pourcentages en rouge, accompagnés de graphiques à barres. Enfin, aucun résultat ne concerne un autre Bill que Clinton, ni un autre Clinton que Bill. Cela s'explique par l'importance que nous accordons à la proximité des mots. Bien entendu, un véritable test de la qualité d'un moteur de recherche nécessiterait une étude approfondie des utilisateurs ou une analyse des résultats, ce que nous ne pouvons pas faire ici. Nous invitons donc le lecteur à tester Google par lui-même sur http://google.stanford.edu.

Outre la qualité de la recherche, Google est conçu pour s'adapter de manière rentable à la taille du Web, à mesure qu'il se développe. L'un des aspects de cette optimisation est l'optimisation de l'espace de stockage. Le tableau 1 présente une analyse détaillée de certaines statistiques et des besoins de stockage de Google.

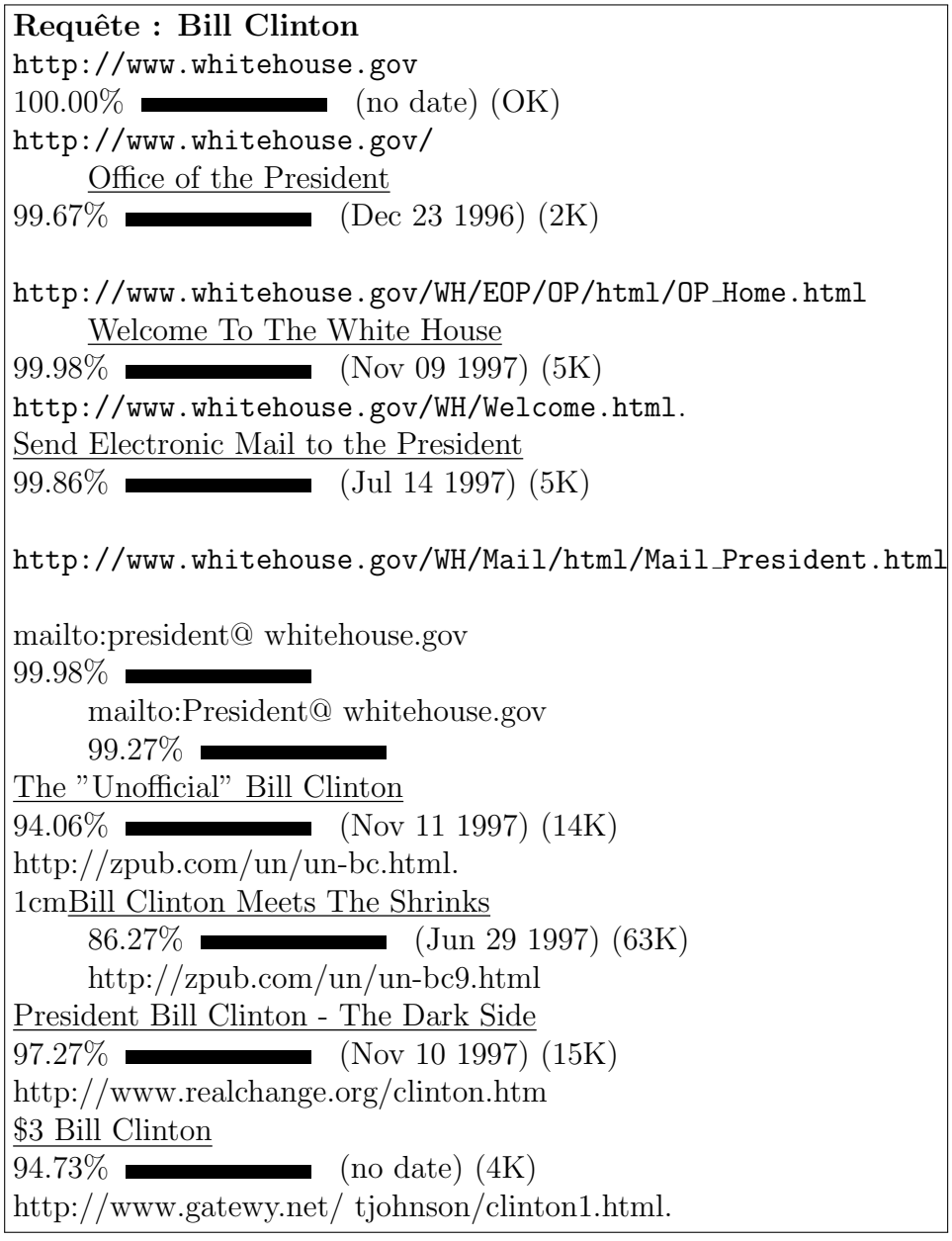


FIG. 2. Exemples de résultats obtenus avec Google.

Il est important pour un moteur de recherche d'explorer et d'indexer efficacement. Cela permet de maintenir les informations à jour et de tester les modifications majeures du système relativement rapidement. Au total, il a fallu environ 9 jours pour télécharger les 26 millions de pages (erreurs comprises). Cependant, une fois le système opérationnel, le téléchargement a été beaucoup plus rapide, les 11 derniers millions de pages ayant été téléchargés en seulement 63 heures, soit une

moyenne d'un peu plus de 4 millions de pages par jour, soit 48,5 pages par seconde. L'indexeur tourne à environ 54 pages par seconde. Les trieurs peuvent être exécutés entièrement en parallèle; avec quatre machines, le processus de tri prend environ 24 heures.

Table 1

Statistics

Statistiques concernant la mémoire	
Taille totale des pages traitées	147.8 GB
Stockage compressé	53.5 GB
Index inversé court	4.1 GB
Index inversé complet	37.2 GB
Lexique	293 MB
Données temporaires d'ancrage (non en totalité)	6.6 GB
Index des documents incluant les données de taille variable	9.7 GB
Base de données des liens	3.9 GB
Total sans stockage	55.2 GB
Total avec stockage	108.7 GB

Statistiques concernant les pages Web	
Nombre de pages Web traitées	24 million
Nombre d'URL visitées	76.5 million
Nombre d'adresses mail	1.7 million
Nombre d'erreurs 404	1.6 million

L'amélioration des performances de recherche n'a pas été l'objectif principal de nos recherches jusqu'à présent. La version actuelle de Google répond à la plupart des requêtes en 1 à 10 secondes. Ce temps est principalement dû aux E/S (entrées/sorties) disque via NFS (puisque nos disques sont répartis sur plusieurs machines). De plus, Google ne dispose pas de nombreuses optimisations courantes utilisées pour accélérer les systèmes de recherche d'informations, telles que la mise en cache des requêtes, les sous-index sur des termes courants et d'autres optimisations courantes.

Nous prévoyons d'accélérer considérablement Google à l'avenir. Le tableau 2 présente quelques exemples de temps de requête pour la version actuelle de Google.

6. Conclusions

Google est conçu pour être un moteur de recherche évolutif. Son objectif principal est de fournir des résultats de recherche de haute qualité sur un Web en pleine expansion. Google utilise plusieurs techniques pour améliorer la qualité de la recherche, notamment le classement des pages, le texte d'ancrage et les informations de proximité. De plus, Google est une architecture complète permettant de collecter, d'indexer et d'exécuter des requêtes de recherche sur les pages Web.

6.1. Travaux futurs

Un moteur de recherche Web à grande échelle est un système complexe et il reste encore beaucoup à faire. Nos objectifs immédiats sont d'améliorer l'efficacité de la recherche et d'atteindre environ 100 millions de pages Web. Parmi les améliorations simples à apporter à l'efficacité, on peut citer la mise en cache des requêtes, l'allocation intelligente de disque et les sous-index. Les mises à jour constituent un autre domaine nécessitant de nombreuses recherches. Nous devons disposer d'algorithmes intelligents pour déterminer quelles anciennes pages Web doivent être réexplorées et quelles nouvelles doivent l'être. Des travaux en ce sens ont été réalisés dans [2]. Un domaine de recherche prometteur est l'utilisation de caches proxy pour la création de bases de données de recherche, car ils sont pilotés par la demande. Nous prévoyons d'ajouter des fonctionnalités simples prises en charge par les moteurs de recherche commerciaux, comme les opérateurs booléens, la négation et la recherche de radicaux. Cependant, d'autres fonctionnalités commencent tout juste à être explorées, comme le retour d'information sur la pertinence et le clustering (Google prend actuellement en charge un clustering simple basé sur le nom d'hôte). Nous prévoyons également de prendre en charge le contexte utilisateur (comme sa localisation) et la synthèse des résultats. Nous travaillons également à étendre l'utilisation de la structure et du texte des liens. Des expériences simples indiquent que le PageRank peut être personnalisé en augmentant le poids de la page d'accueil ou des favoris d'un utilisateur. Quant au texte des liens, nous expérimentons l'utilisation de texte entourant les liens en plus du texte du lien lui-même. Un moteur de recherche Web est un environnement très riche pour les idées de recherche. Nous en avons beaucoup trop pour les énumérer ici, nous ne prévoyons donc pas que cette section "Travaux futurs" devienne beaucoup plus courte dans un avenir proche.

Table 2
Temps de recherche

Requête	Requête initiale		Même requête répétée (10	caches les plus fréquents)
	Temps CPU (s)	Temps total (s)	Temps CPU (s)	Temps total (s)
Al Gore	0.09	2.13	0.06	0.06
vice president	1.77	3.84	1.66	1.80
hard disks	0.25	4.86	0.20	0.24
search engines	1.31	9.63	1.16	1.16

6.2. Recherche de haute qualité

Le plus gros problème auquel sont confrontés les utilisateurs des moteurs de recherche Web aujourd'hui est la qualité des résultats qu'ils obtiennent. Bien que les résultats soient souvent amusants et élargissent les horizons des utilisateurs, ils sont souvent frustrants et prennent un temps précieux. Par exemple, le meilleur résultat pour une recherche de "Bill Clinton" sur l'un des moteurs de recherche commerciaux les plus populaires était "*la blague du jour sur Bill Clinton : 14 avril 1997*" [7]. Google est conçu pour fournir une recherche de meilleure qualité afin que, à mesure que le Web continue de croître rapidement, les informations puissent être trouvées facilement. Pour ce faire, Google utilise abondamment les informations hypertextuelles, composées de la structure et du texte des liens (ancres). Google utilise également des informations de proximité et de police. Bien que l'évaluation d'un moteur de recherche soit difficile, nous avons subjectivement constaté

[7] *lien 404 :* `http://www.io.com/ cjburke/clinton/970414.html`

que Google renvoie des résultats de recherche de meilleure qualité que les moteurs de recherche commerciaux actuels. L'analyse de la structure des liens via le PageRank permet à Google d'évaluer la qualité des pages Web. L'utilisation du texte du lien comme description de ce vers quoi le lien pointe aide le moteur de recherche à renvoyer des résultats pertinents (et dans une certaine mesure de haute qualité). Enfin, l'utilisation d'informations de proximité contribue à accroître considérablement la pertinence pour de nombreuses requêtes.

6.3. Architecture évolutive

Outre la qualité de la recherche qu'il permet, Google est conçu pour évoluer. Il doit être efficace dans l'espace et dans le temps, et les facteurs constants sont très importants lorsqu'on traite l'ensemble du Web. Lors de la mise en œuvre de Google, nous avons constaté des goulots d'étranglement au niveau du processeur, de l'accès mémoire, de la capacité de la mémoire, des recherches sur le disque, du débit du disque, de la capacité du disque et des E/S (entrées/sorties) réseau. Google a évolué pour surmonter plusieurs de ces goulots d'étranglement lors de diverses améliorations. Les principales structures de données de Google utilisent efficacement l'espace de stockage disponible. De plus, les opérations d'exploration, d'indexation et de tri sont suffisamment efficaces pour permettre de créer un index d'une partie substantielle du Web (24 millions de pages) en moins d'une semaine. Nous prévoyons de créer un index de 100 millions de pages en moins d'un mois.

6.4. Un outil de recherche

Google est un outil de recherche. Les données collectées par Google ont déjà donné lieu à de nombreux articles soumis à des conférences et à bien d'autres à venir. Des recherches récentes, telles que [11], ont mis en évidence un certain nombre de limites aux requêtes sur le Web, auxquelles il est possible de répondre sans accès local au Web. Cela signifie que Google (ou un système similaire) est non seulement un outil de recherche précieux, mais aussi indispensable pour un large éventail d'applications. Nous espérons que Google deviendra une ressource précieuse pour les chercheurs du monde entier et qu'il ouvrira la voie à la prochaine génération de moteurs de recherche.

Remerciements : Scott Hassan et Alan Steremberg ont joué un rôle essentiel dans le développement de Google. Leurs contributions talentueuses sont irremplaçables et les auteurs leur doivent une grande gratitude. Nous tenons également à remercier Hector Garcia-Molina, Rajeev Motwani, Jeff UII-man et Terry Winograd, ainsi que l'ensemble du groupe WebBase, pour leur soutien et leurs discussions éclairées. Enfin, nous tenons à remercier nos donateurs d'équipement, IBM, Intel et Sun, ainsi que nos bailleurs de fonds, pour leur généreux soutien. Les recherches décrites ici ont été menées dans le cadre du projet de bibliothèque numérique intégrée de Stanford, soutenu par la National Science Foundation dans le cadre de l'accord de coopération IRI-9411306. Le financement de cet accord de coopération est également assuré par la DARPA et la NASA, ainsi que par Interval Research et les partenaires industriels du projet de bibliothèques numériques de Stanford.

Références

[1] S. Abiteboul, V. Vianu, Queries and computation on the Web, *Proceedings of the International Conference on Database Theory*. Delphi. Greece, 1997.

[2] J. Cho, H. Garcia-Molina, L. Page, Efficient crawling through URL ordering, *Proc. of the 7th International World Wide Web Conference (WWW 98), Brisbane. Australia.* April 14-18, 1998; also Comput. Networks ISDN Systems. 30(1-7): 161-172 (this volume).

[3] J. Kleinberg. Authoritative sources in a hyperlinked environment, Proc. *ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[4] M. Marchiori. The quest for correct information on the Web: hyper search engines, *Proc. of the 6th International WWW Conference (WWW 97)*. Santa Clara. USA, April 7-11, 1997.

[5] Mauldin. M.L., Lycos design choices in an Internet search service, IEEE Expert Interview, http://www.computer.org/p ubs/expert/1997/trends/x1008/mauldin.htm.

[6] O.A. McBryan, GENVL, and WWWW: tools for taming the Web, *Proc. of the Ist International Conference on the World Wide Web*, CERN, Geneva, Switzerland, May 25-27. 1994, http://www.cs.colorado.edu/home/mcbryan/mypapers /www94.ps.

[7] L. Page. S. Brin. R. Motwani and T. Winograd, The PageRank citation ranking: bringing order to the Web. Manuscript en cours de rédaction, http://google.stanford.edu/ backru b/pageranksub.ps.

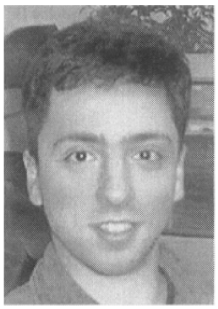
[8] B. Pinkerton, Finding what people want: experiences with the WebCrawler, *Proc. of the 2nd International WWW Conference*, Chicago, USA, October 17-20, 1994, http://info.webcrawler.com/bp/WWW94.html.

[9] E. Spertus. ParaSite: mining structural information on the Web, *Proc. of the 6th International WWW Conference (WWW 97)*. Santa Clara, USA, April 7-11. 1997.

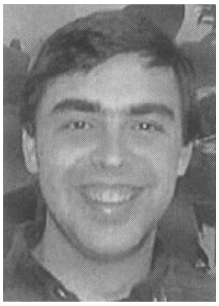
[10] D.K. Harman and E.M. Voorhees (Eds.). *Proceedings of the Fifth Text Retrieval Conference (TREC-5)*, Gaithers-burg. Maryland, November 20-22, 1996, Department of Commerce. National Institute of Standards and Technology. 1996; full text at http://trec.nist.gov/.

[11] I.H Witten, A. Moffat. and T.C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*, Van Nostrand Reinhold, New York, NY, 1994.

[12] R. Weiss. B. Velez. M.A. Sheldon, C. Manprempre, P. Szilagyi, A. Duda. and D. K. Gifford, HyPursuit: a hierarchical network search engine that exploits content-link hypertext clustering, *Proc. of the 7th ACM Conference on Hypertext*, New York, 1996.



Sergey Brin a obtenu sa licence en mathématiques et informatique à l'Université du Maryland à College Park en 1993. Il est actuellement doctorant en informatique à l'Université Stanford, où il a obtenu sa maîtrise en 1995. Il est titulaire d'une bourse d'études supérieures de la National Science Foundation. Ses recherches portent sur les moteurs de recherche, l'extraction d'informations à partir de sources non structurées et l'exploration de données à partir de vastes collections de textes et de données scientifiques.



Lawrence Page est né à East Lansing, dans le Michigan, et a obtenu une licence en génie informatique à l'Université du Michigan à Ann Arbor en 1995. Il est actuellement doctorant en informatique à l'Université Stanford. Ses recherches portent notamment sur la structure des liens du Web, l'interaction homme-machine, les moteurs de recherche, l'évolutivité des interfaces d'accès à l'information et l'exploration de données personnelles.