

HTTP/2 : quel impact en SEO ?



**Par Guillaume et
Sylvain Peyronnet**

Domaine :	Recherche	Référencement
Niveau :	Pour tous	Avancé

Le protocole http est à la base du Web et régit les échanges entre navigateur et serveur. Sa version 1.1, datant de quinze ans, commençait à dater quelque peu au vu de l'évolution actuelle du Web. Sa nouvelle mouture, baptisée http/2, a donc vocation à accélérer les échanges de données sur le réseau. Quelles en sont ses principales caractéristiques ? Et quel en sera l'impact sur le SEO ? Réponses...

Une charge historique forte

Le protocole http a été créé en 1990. En 99, il a connu une mise à jour notable, en passant en version 1.1. Depuis, s'il rend au jour le jour de formidables services en permettant notamment à tout un chacun de consulter des sites web en tous genres, il n'a pas connu d'évolution notable. Rien, donc, depuis 15 ans...

Le http a été conçu comme un protocole qui permet les échanges entre un client (le navigateur web) et un serveur (le site web). Le processus est le suivant : lorsque l'on navigue sur le web, qu'on accède, par exemple, à une page web, une requête http est envoyée au serveur hébergeant le site web correspondant, en lui demandant de fournir la ressource *ad hoc*. Le serveur crée le fichier html demandé, et envoie en retour une réponse http.

Le fichier récupéré contient le code source de la page web consultée, et de nombreuses références à des ressources annexes (en moyenne une centaine par pages) qui permettent de rendre la page plus attrayante : fichiers CSS, images, Javascript, etc.

Pour récupérer ces ressources, le navigateur web envoie une nouvelle requête http. Le serveur web la reçoit et si tout se passe bien (la ressource existe bien à l'emplacement indiqué) il fournit une réponse http la contenant.

Pour chaque ressource utile à une page, on a donc une requête http à faire. Chaque requête à envoyer au serveur web engendre un délai de quelques centièmes de secondes, du fait de la distance entre le client et le serveur. Ce n'est certes pas grand chose, mais finalement, quand on parle d'une centaine de ressources par page sur le Web récent, l'addition commence à être bien salée et le tout peut devenir très lent à l'usage.

Google entre dans la danse

La consultation des sites web est ainsi, par essence, plutôt lente. Dès lors, lorsque l'on est un gros acteur du web, qu'on parcourt des millions, voire des milliards de pages web chaque jour, on perd beaucoup de temps à récupérer les ressources. Si l'on était capable de diviser par deux la vitesse de chargement des pages, on pourrait parcourir le Web en deux fois moins de temps. Et quand le temps de parcours

est proportionnel aux coûts, on a un intérêt non négligeable à inciter tous les webmasters à faire des efforts.

C'est sans doute de ce constat que sont parties les équipes de Google quand il a été décidé de faire de la vitesse de chargement un critère de classement pour le moteur de recherche (cf. le blog de Google :

<http://googlewebmastercentral.blogspot.no/2010/04/using-site-speed-in-web-search-ranking.html>). Le moteur s'est donc fait un devoir, dans un premier temps, de pousser les webmasters à optimiser leurs sites pour qu'ils soient plus rapides (en donnant accès à des bonnes pratiques et des outils de monitoring comme [Google PageSpeed Insights](#), ou encore des plugins pour des serveurs web comme [Nginx](#) et [Apache](#)). Mais, plus ambitieux, Google a commencé à plancher sur une amélioration de http, sous le nom de SPDY (à prononcer « speedy »). L'idée étant de corriger toutes les caractéristiques de http 1.1 qui sont bloquantes pour la vitesse.

La sécurité pour tous

Encore plus ambitieux, Google décide de faire de SPDY un protocole uniquement disponible en format sécurisé (https/ssl).

C'est assez contraire à l'idée de base, qui est d'accélérer les temps de chargement, puisqu'ajouter une étape d'identification là où il n'y en avait pas, entraîne nécessairement un délai supplémentaire.

Par ailleurs, Google, pour promouvoir la mise en place du https sur les sites, a proclamé dernièrement que cette sécurisation devenait un critère de

positionnement pour son moteur (<http://googlewebmastercentral.blogspot.fr/2014/08/https-as-ranking-signal.html>). On n'a pour l'instant pas vraiment de preuve que cela soit vrai, mais, pour prévoir le futur, mieux vaut garder ce paramètre à l'esprit.

Apparition de http/2

SPDY, grâce à la publication de tests de performance très positifs (cf. <http://www.chromium.org/spdy/spdy-whitepaper>), a gagné de beaux succès d'estime. SPDY accélère sensiblement le chargement des pages, et, même si on ne le voit pas dans le tableau ci-dessous, accélère d'autant plus le chargement que la bande-passante disponible est élevée.

	Temps de chargement moyen (ms)	Accélération
http 1.1	2348	-
SPDY Connexion TCP (moyenne)	1016	57%
SPDY Connexion SSL (moyenne)	1073	54%

(tableau d'évolution de la performance de SPDY face à http 1.1 – comparatif effectué sur 25 sites du TOP 100 – 4 Mb/s)

Plusieurs serveurs web, ainsi que les versions récentes des principaux navigateurs web ont alors décidé d'intégrer les spécifications de SPDY, si bien qu'il est déjà possible depuis quelques mois de déployer sans sourciller SPDY sur son site.

Tout laisse penser que SPDY est déjà un succès. C'est exact, mais la standardisation est la seule voie vers le réel succès.

Ainsi, c'est assez naturellement que http/2, le futur successeur de http 1.1, a pris comme modèle SPDY. Si bien que très récemment, le 9 février dernier, les équipes développant SPDY ont annoncées arrêter leurs travaux pour se consacrer à http/2 (cf. <http://blog.chromium.org/2015/02/hello-http2-goodbye-spdy-http-is-9.html>).

On peut donc en être sûr, http/2 est plus que jamais d'actualité. Il va falloir prochainement faire le nécessaire pour que nos sites le supportent. Et c'est une excellente nouvelle, car cette future version de http/2 a des atouts incroyables !

Les promesses de http/2

Http/2, c'est avant tout la promesse d'un protocole plus rapide, axé sur trois caractéristiques principales :

- Compression des entêtes http ;
- Push serveur des ressources ;
- Chargement des ressources en parallèle, via une seule connexion TCP.

Compression des entêtes http

Lorsque l'on demande une ressource web en http 1.1, elle est renvoyée avec une entête (URL de la ressource, code http, cookie, etc.). Si cette entête n'est généralement longue que de quelques lignes, et requiert donc peu de temps à télécharger, elle est présente sur toutes les ressources. Si une page web fait appel à 100 ressources distinctes, ce

sont 100 entêtes qu'il faut charger, et donc une multiplication notable du temps de chargement.

En compressant les entêtes, qui de plus sont purement textuelles, et donc subissent avec plaisir la cure d'amaigrissement, le http/2 gagne finalement beaucoup en vitesse.

Push serveur des ressources

En http 1.1, lorsque l'on souhaite afficher une page web, on commence par récupérer le code source html, qui est aussitôt analysé par le navigateur. Ce dernier demande alors les différentes ressources qu'il a repéré. Une première requête http est donc effectuée, et les autres requêtes sont faites seulement après analyse, ce qui entraîne nécessairement un délai.

En http/2, le serveur est capable de « pousser » les ressources (« push ») vers le cache navigateur avant même que ce dernier ne soit conscient d'en avoir besoin. Tout devient plus rapide puisqu'il devient inutile pour le navigateur de réclamer les ressources qu'il a déjà reçu.

Chargement des ressources en parallèle, via une seule connexion TCP

En http 1.1, nous l'avons vu, une ressource à charger équivaut à un échange complet avec le serveur web (demande de la ressource puis réception de la ressource). Puisqu'une page web est généralement constituée de plusieurs dizaines de ressources, la solution la plus facile est de penser parallélisation. Pourquoi le navigateur web ne pourrait-il faire qu'une seule requête à la fois ? Pourquoi ne pas en

faire 2, 4, voire 6 à la fois ? Si en 90 on hésitait à surcharger ainsi les échanges Internet du fait des débits de qualités très variées et certainement insuffisants, pourquoi se priver dès lors que les « tuyaux » le permettent ?

Effectivement, paralléliser les requêtes est une bonne idée. C'est ce que fait un navigateur web depuis de nombreuses années maintenant. Et cela permet réellement d'accélérer la vitesse de chargement des sites.

Mais, étrangement, les navigateurs ne se sont jamais vraiment mis à la page : le nombre de requêtes parallèles (c'est-à-dire le nombre de connexions) ne dépend pas de la qualité de la ligne de l'internaute, mais simplement d'une valeur arbitraire. Généralement de 4 à 6 requêtes en simultané pour un domaine ciblé. C'est donc plus rapide, mais toujours pas la panacée.

Les développeurs *front-end* ont alors pris le problème à bras le corps, et ont inventé une astuce toute simple : puisque la limitation ne concerne qu'un domaine particulier, pourquoi ne pas charger les ressources à partir de plusieurs domaines ?

Par exemple, le site www.example.org place ses fichiers Javascript sur js.example.org, ses feuilles de styles CSS sur css.example.org et sa galerie d'images sur images.example.org. Dès lors, on considère quatre domaines différents (pour être plus précis, c'est le sous-domaine qui compte au delà du nom de domaine), et on peut donc faire, par exemple quatre fois plus de requêtes que d'habitude en même temps. C'est-à-dire aller quatre fois plus vite si le réseau le permet. CQFD...

Intéressant, certes, mais tellement peu pratique... Il s'agit ici de bricolage, certes bien pensé, mais de bricolage tout de même. Sur le Web, ce type d'astuce est loin d'être généralisé, et on constate donc un gros manque à gagner pour la vitesse sur le Web, en général.

C'est là qu'intervient http/2. Plutôt que de demander les ressources une par une, le navigateur envoie une liste complète des ressources qu'il souhaite récupérer. De son côté, le serveur web, après étude de la liste de ressources, commence à les renvoyer dès que possible, en utilisant la bande-passante disponible, en parallèle. On peut ainsi récupérer plusieurs images d'un seul coup, plutôt que d'attendre que la première se charge avant de récupérer la seconde. Le gain de temps peut se révéler réellement appréciable.

De fait, cela permet de mettre fin au problème de la ressource bloquante : en http 1.1, si la première ressource à charger est un fichier JS et que celui-ci pèse plusieurs Mo, pendant ce temps-là, rien d'autre ne se passait. En http/2, l'obstacle disparaît (ce qui ne doit pas être une autorisation implicite de faire d'énormes fichiers JS, bien sûr).

L'autre avantage de http/2 est de faire passer tous les échanges *via* une unique connexion. On ne crée ainsi qu'un seul tunnel entre un serveur web et un client (bien sûr, ce tunnel peut accueillir une route avec plusieurs files - c'est le concept même de la parallélisation). Or, dans une montagne, on ne peut pas creuser une infinité de tunnels. En http/2, un serveur peut donc accueillir un nombre de clients égal au nombre de tunnels que l'on peut creuser.

En http 1.1, un même client, à cause des astuces d'optimisations de vitesse, utilise à lui seul plusieurs tunnels, prenant potentiellement la place d'autres clients. Avec la même montagne - le même serveur web - on peut de fait accueillir plus de visiteurs en http/2.

http/2 et sécurité

Nous avons vu auparavant que SPDY et Google proposaient l'idée d'un web entièrement en https. Si cette option est formidable pour améliorer la sécurité des échanges, il s'agit également d'un frein à la vitesse de chargement des sites web.

En effet, avec SPDY, davantage d'échanges entre clients et serveurs sont créés, puisqu'il faut des échanges de clés pour en assurer l'authentification. C'est de fait un délai supplémentaire.

De même, les échanges étant codés, il faut à un moment donné que le serveur web et le client les cryptent et les décryptent. Cela représente *a priori* un délai supplémentaire dans le traitement, mais les experts s'accordent pour dire que sur une machine moderne, ce point est quasiment transparent.

Si le https fait partie de l'ADN de SPDY, c'est moins vrai pour http/2, au moins dans sa version actuellement à l'étude. On ne sait pas encore si le fait de mettre en place des certificats SSL sera une nécessité ou si l'on pourra s'en passer.

Il faut bien dire que pour de nombreux types de sites web, ceux où les informations ne sont pas sensibles,

proposer des échanges sécurisés n'est pas nécessairement pertinent. Si cela se fait au prix d'une baisse des performances, ou d'une amélioration moindre, on peut légitimement se poser quelques questions...

Cependant, l'écart de performance généralement constaté entre http 1.1 et http/2 est tel que, même en ajoutant une couche SSL, on peut parier que le gain sera au rendez-vous, au prix peut-être d'un rendement moins optimal.

Impact sur le référencement web

Nous l'avons vu, http/2 est avant tout une évolution orientée performance. Il représente un protocole formidable pour améliorer notablement la vitesse sur l'ensemble du Web.

Dans les faits, les sites déjà extrêmement optimisés pour le http 1.1, qui utilisent les meilleures bonnes pratiques ne doivent pas s'attendre à des changements spectaculaires. Mais pour tous les autres sites et pour le temps de chargement moyen sur le Web, on peut attendre de beaux résultats, voire quelques miracles, et une réelle avancée.

Ainsi, comme Google donne certainement - en tout cas, il s'agit là de son discours - une prime à la performance (crawl de façon certaine et peut-être positionnement), on a tout intérêt à mettre en place dès que possible le http/2. Pourquoi « dès que possible » ? Tout simplement parce qu'une fois que http/2 sera partout, il n'y aura plus grande différence entre les différents sites. Si vous vous lancez au plus vite, vous pouvez éventuellement espérer une petite prime.

Mais honnêtement, il ne faudra pas s'attendre à un grand bond, puisque tout laisse penser que la « vitesse » considérée par Google est en réalité le TTFB (*Time To First Byte*), c'est à dire le temps nécessaire au navigateur pour recevoir le premier bit de la réponse du serveur web. Cela inclut notamment le temps pour le serveur de fabriquer la page (si celle-ci est dynamique) mais pas du tout les ressources externes. La vitesse pour Google, ce serait plus la vitesse de réaction du serveur que celle du site à proprement parler.

Preuve en est l'étude de corrélation faite par Zoompf et publié chez Moz (cf. <http://moz.com/blog/how-website-speed-actually-impacts-search-ranking>) : il n'existe aucune corrélation entre le temps de chargement « classique » d'une page web et son positionnement dans Google, alors qu'il existe une corrélation avec le TTFB. Il s'agit donc d'une piste à suivre et à actionner, même si on n'est sûr de rien car corrélation ne vaut pas causalité.

Le http/2 ne sera pas un miracle pour les référenceurs, sauf peut-être pour ceux qui prennent le train plus vite que les autres – et encore si la prime va plus loin que le simple TTFB. Et pourtant il sera un véritable eldorado pour eux.

En effet, si on met de côté le référencement impliquant du pur positionnement, on peut élargir le concept à la visibilité, mais surtout à la conversion. Un client heureux n'est-il pas le plus souvent un client qui convertit mieux plutôt qu'un client qui se positionne dans les SERP ?

Dans cette optique, améliorer la performance de son site web est

primordial : on sait - les études sont nombreuses à ce sujet – que la vitesse permet d'améliorer la conversion, le nombre de transactions, de pages vues, les paniers moyens, etc.

Si les visiteurs sont plus enclins à rester sur votre site, ils retourneront également moins vite sur le moteur pour compléter leur recherche, ce qui permettra de réduire le taux de rebond (*pogosticking*), et devrait donc satisfaire Google, par effet de bord, pour tout ce qui implique le volet d'étude et d'ajustement comportemental du moteur.

Mais si vous avez un site web déjà très optimisé, utilisant toutes les bonnes pratiques, donc déjà très rapide, le passage au http/2 se fera bien entendu de façon obligatoire, mais rien ne presse.

Si en plus vous ne faites pas de https, vous avez peut-être tout intérêt à attendre que les spécifications deviennent plus concrètes : faire le grand saut vers le tout SSL n'est pas toujours de tout repos...

Conclusion

Le http/2 se fait attendre depuis des années, mais il semble ne pas vouloir décevoir. On peut espérer – au moins partiellement - la fin des bidouilles pour les développeurs *front-end*, tandis que pour le référenceur, l'impact est finalement bien plus marginal.

On peut aussi s'attendre à une certaine prime si l'on est parmi les premiers à adopter le futur protocole. Mais de là à penser faire la différence sur les

positionnements,
malheureusement utopique.

c'est

sans trop d'efforts, pourquoi faudrait-il s'en
priver ?

En revanche, au niveau de l'expérience
utilisateur, http/2 représente une
opportunité temporaire à saisir dès que
possible. Si on peut accélérer de 18 à 60 %
la vitesse d'un site via http/2, et en plus



Guillaume Peyronnet est gérant de
Nalrem Médias. **Sylvain Peyronnet** est co-
fondateur et responsable des ix-labs, un
laboratoire de recherche privé. Ensemble,
ils font des formations, pour en savoir plus
: <http://www.peyronnet.eu/blog/>