

Qu'est-ce que Google Wave ?

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 26 octobre 2009. Dernière mise à jour le 25 août 2010

<https://www.bortzmeyer.org/googlewave.html>

N'espérez pas trouver ici une réponse définitive à cette question. Je débute, moi aussi, et je suis loin d'avoir tout compris. Et ce que j'ai compris, je l'ai peut-être compris de travers. Mais, au moins, grâce au Web, je peux partager mon ignorance.

Dans un grand concert d'agitation médiatique, Google a lancé les premières invitations pour participer au bêta-test de Google Wave. (Le projet a depuis été abandonné par Google <<http://googleblog.blogspot.com/2010/08/update-on-google-wave.html>>.) Cette technique marketing permet à la fois de donner aux destinataires de l'invitation l'impression flatteuse d'être des privilégiés, et à l'infrastructure technique et humaine de Google de ne pas s'écrouler sous une charge subite. Mon identificateur Google Wave est `bortzmeyer@googlewave.com` pour ceux qui veulent essayer avec moi. Attention, même si ça y ressemble, ce n'est **pas** une adresse de courrier.

D'abord, à quoi sert Google Wave? Tout le monde bute sur cette question. Google ne fournit que des vidéos de démonstration, jolies mais peu claires, et des documents techniques touffus. Le texte écrit est apparemment réservé, chez Google, aux techniciens, les autres n'ont qu'à regarder les images. Sur les jolies vidéos, on voit plusieurs personnes éditer un document multimédia partagé via l'Internet. Soit, et alors? Google Docs, de la même entreprise, en fait autant depuis longtemps (ou bien, en plus libre, PiratePad <<http://piratepad.net/>>/EtherPad <<http://code.google.com/p/etherpad/>>). On voit aussi deux personnes communiquer en « temps réel ». En quoi est-ce différent de la traditionnelle messagerie instantanée telle qu'elle se fait avec XMPP ou, depuis encore plus longtemps, avec IRC? Ou bien Google Wave serait-il un concurrent du courrier? On a bien vu des gens écrire qu'il éviterait le spam, en raison du stockage du document chez l'émetteur (idée tout à fait fausse, comme le montre bien la présence d'un bouton de signalement de spam sur le banc de test...)

Les points importants de Google Wave, semble t-il, sont justement ceux qui ne sont pas dans les vidéos. Par rapport à Google Docs, la grande différence est que Google Wave n'est pas (ou, plus exactement, ne **sera** pas) un service centralisé. Avec Google Docs, vous abandonnez tout à Google. Vos documents sont hébergés chez Google qui peut en faire ce qu'il veut. Google Docs était donc un grand retour en arrière par rapport à ce que permet l'Internet. Google Wave, au contraire, est **fédératif** (comme le courrier électronique). Chacun peut (au moins en théorie) avoir son serveur Wave chez lui et créer des identités à lui, ne se terminant pas par `@googlewave.com`.

Sur cette base, de nombreuses applications semblent possibles. La plus évidente est la création en commun de documents. Aujourd'hui, pour un travail en groupe, il y a deux solutions :

- Celle qui marche bien : mettre un document texte dans un VCS, un logiciel de gestion des versions, et permettre à plusieurs personnes, dispersées sur l'Internet, de le modifier. Tous les VCS permettent de configurer des droits de lecture seule ou bien de lecture et d'écriture. Tous les VCS permettent de **fusionner** les modifications réalisées indépendamment par deux auteurs (quitte à laisser une partie du travail de fusion à l'éditeur, dans le cas où les modifications sont inconciliables). Les VCS peuvent être centralisés (un dépôt unique) ou pas. Tous les VCS, surtout les plus récents, disposent d'outils pour aider au travail en groupe, par exemple en envoyant un message à tous les participants à chaque modification, en fournissant un flux de syndication des modifications ou en créant automatiquement un identificateur par version permettant aux participants, dans les discussions, de se référer de manière non ambiguë à une version spécifique. Cette méthode de travail est très largement utilisée par les informaticiens et a fait ses preuves, non seulement pour le code source mais aussi pour des textes, écrits par exemple dans les formats ouverts <<https://www.bortzmeyer.org/formats-ouverts.html>> comme LaTeX ou Docbook. Mais elle est mal adaptée au multimédia (la fusion n'est intelligente que pour les formats non-binaires). Et, pour le texte, elle impose souvent des contraintes sur l'environnement local (par exemple, avec LaTeX, que tout le monde aie les mêmes paquets...)
- La seconde méthode est celle couramment utilisée en entreprise par les gens qui n'ont pas de culture informatique. Elle consiste à envoyer d'énormes documents issus d'obésiciels WYSIWYG entre tous les participants. Cette méthode gaspille beaucoup de ressources réseau (en raison de la taille des documents). Avec les versions récentes de logiciels comme OpenOffice, la fusion entre des modifications réalisées séparément par deux participants est possible quoique pénible (et cette possibilité ne semble guère utilisée en pratique). Il est très difficile de savoir où en est le projet et quelle est la version qui fait autorité. Si la liste des participants change, il est difficile de s'adapter. Certains outils existent pour faciliter le suivi des modifications (comme la fonction de « Suivi des modifications » dans OpenOffice).
- (Une troisième méthode existe, purement centralisée, c'est celle de Google Docs, qui oblige tout le monde à être connecté au même service. Si tous les participants appartiennent à la même organisation, elle est fréquemment utilisée avec un dépôt de fichiers local à l'organisation.)

Comme aucune de ces méthodes n'est idéale, et que ce scénario est très fréquent dans tous les environnements de travail, il existe donc une bonne raison de tester des nouvelles méthodes. Google Wave est l'une de celles-ci, comme l'explique bien Daniel Tenner dans « *What problems does Google Wave solve?* » <<http://danieltenner.com/posts/0012-google-wave.html>>.

Google Wave permet en effet d'avoir une copie faisant autorité du document (le document, la "wave", l'« onde », a en effet un serveur Wave de référence, où elle est conservée), tout en permettant les modifications faites par divers participants, y compris via leurs propres serveurs Wave. Par contre, Google Wave ne semble pas permettre de travail en mode déconnecté du réseau. Comme d'autres applications, il s'appuie sur la légende comme quoi nous serons, dans un futur inspecifié, connectés en permanence.

Une des difficultés à évaluer Google Wave est la différence entre la théorie et la pratique. Ainsi, en théorie, Google Wave n'impose aucune dépendance vis-à-vis de Google. En pratique, il n'y avait au début qu'un seul serveur Wave et surtout qu'un seul service d'hébergement, celui de Google ("*google-wave.com*"). Depuis, des annonces diverses ont été faites (par exemple que Novell intégrait le protocole Wave dans ses produits <<http://www.novell.com/news/press/novell-unveils-real-time-collaboration>>, comme « Pulse », qui a été montré en janvier 2010 <<http://www.zdnet.com.au/news/software/soa/Novell-pulses-nicely-with-Google-Wave/0,130061733,339300448,00.htm>>), allant dans le sens d'une réalisation des promesses de Wave comme système fédératif.

Voici donc ce que j'ai compris de l'utilité de Google Wave et de l'utilisation qu'on peut en faire. Maintenant, je vais devoir être un peu plus technique pour développer d'autres points.

Google Wave repose sur plusieurs protocoles, tous ouverts et publics, ce qui est un de ses principaux avantages. Ainsi, le protocole entre serveurs Wave (le protocole de fédération) est une extension du protocole XMPP de messagerie instantanée, protocole normalisé dans le RFC 6120¹. XMPP a un mécanisme normalisé d'extension, XEP 114 <<http://xmpp.org/extensions/xep-0114.html>> et c'est dans le cadre de ce mécanisme que Google a spécifié les extensions Wave <<http://www.waveprotocol.org/draft-protocol-specs/draft-protocol-spec>>.

Ce protocole entre serveurs Wave dépend d'une technique très complexe pour gérer les modifications indépendantes du document : la transformation d'opérations. Elle dira quelque chose à tous ceux qui ont transpiré sur des techniques analogues pour les VCS comme par exemple la théorie des patches <<http://darcs.net/manual/node9.html>> de darcs. Pour un point de départ, voyez l'excellent résumé de Wikipédia sur la fusion des modifications et pour une vision plus complète de cette transformation « *"The Secret Sauce Behind Google Wave"* <http://blogs.gartner.com/ray_valdes/2009/05/31/the-secret-sauce-behind-google-wave/> ». Les extensions (encore!) de Google à cette technique de transformation des opérations sont décrites dans « *"Google Wave Operational Transformation"* <<http://www.waveprotocol.org/whitepapers/operational-transform>> ». (Les utilisateurs de Xwiki noteront qu'il existe un projet similaire <<http://www.pubbs.net/xwiki/200909/96661/>> pour Xwiki.)

Quel est le principe de la transformation d'opérations? Lorsque plusieurs sites modifient le même document de manière non coordonnées, et s'envoient les modifications, on souhaite que le document résultant soit le même sur tous les sites. Or, une mise en œuvre naïve ne va pas produire ce résultat. Supposons seulement deux sites, Alice et Bob. Supposons des documents qui soient composés d'une simple chaîne de caractères, par exemple « abcdef ». Supposons enfin qu'on modélise les changements possibles par seulement deux opérations, `delete(C, N)` qui détruit le caractère C situé au point N, et `insert(C, N)` qui insère le caractère C au point N.

Alice supprime le A initial (`delete('a', 1)`) et Bob ajoute un x en deuxième position (`insert('x', 2)`). Alice a désormais le document « bcdef » et Bob le document « axbcdef ». Ils s'envoient les opérations effectuées, pour se synchroniser. Si chacun effectue bêtement l'opération de l'autre, Alice va se retrouver avec « bxcdef » (elle a inséré le caractère 'x' en seconde position, bien que les caractères aient tous été décalés par la suppression de 'a') et Bob va avoir « xbcdef ». Les deux documents sont désormais différents.

La solution de la transformation des opérations consiste à **transformer** l'opération d'édition avant de l'appliquer. Un algorithme possible, pour cette modélisation, est « la transformation d'un `insert` par rapport à un `delete` se fait en gardant N si le N du `delete` lui était supérieur et en remplaçant N par N-1 si le N du `delete` était inférieur » (notez que, pour le cas où les deux N sont égaux, il faut décider arbitrairement qui gagnera : dans Wave, chaque onde - ou plus exactement chaque ondelette - a un site de référence, qui gagne toujours en cas de conflit).

Avec cet algorithme, l'opération `insert('x', 2)` de Bob devient, lorsqu'elle est transformée par Alice, `insert('x', 1)`. Ainsi, la lettre 'x' est bien insérée au début et les deux documents sont désormais identiques. Bien sûr, les vraies transformations sont bien plus complexes et il existe des tas de façons de les spécifier. L'article de Wikipédia donne une première idée. Par exemple, il faut maintenir une horloge (un compteur) sur chaque site, et indiquer, pour chaque opération transmise, l'« heure » à laquelle elle a été faite pour que les transformations soient appliquées à la bonne version.

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc6120.txt>

Contrairement au protocole entre serveurs, le protocole entre un client Wave et un serveur Wave est très différent et n'est **pas** spécifié à l'heure actuelle (mais des discussions existent http://groups.google.com/group/wave-protocol/browse_thread/thread/184b66dfb3b9a24d). On ne peut donc pas écrire un client Wave standard. Le seul opérationnel aujourd'hui est le client Web de Google <http://wave.google.com/>, qui utilise un protocole nommé GWT (des « clients » comme WaveBoard <http://www.getwaveboard.com/> sont juste des emballages d'un navigateur Web). Il existe aussi des clients expérimentaux comme un mode Emacs <http://code.google.com/p/wave-client-fo>.

Si la variété des protocoles nécessaires pour Wave vous perturbe, vous trouverez de bonnes explications dans des articles comme « *Google Wave Protocols : Clearing the Confusion* » http://gilbane.com/blog/2009/09/google_wave_protocols_clearing_the_confusion.html » (avec un beau dessin), « *Wave's Web of Protocols* » <http://www.cubiclemuses.com/cm/articles/2009/08/09/waves-web-of-protocols/> » ou dans « *Wave Protocol Thoughts* » <http://bitworking.org/news/431/wave-first-thoughts> ». Si vous aimez regardez les paquets qui circulent sur le réseau, [pcapr.net](https://www.bortzmeyer.org/pcapr.html) <https://www.bortzmeyer.org/pcapr.html> a déjà des paquets Wave <http://pcapr.net/browse?q=wave>.

En parlant de logiciel, s'il n'existe pas aujourd'hui d'autres services Wave que celui de Google, notons qu'un logiciel serveur, écrit en Java, existe sous une licence libre. Il est fourni par Google <http://code.google.com/p/wave-protocol/wiki/Installation> et vient avec son propre client <http://code.google.com/p/wave-protocol/wiki/ConsoleClient> (incompatible avec le service de Google, comme indiqué). Comme il fonctionne dans le cadre du XEP 114, il a besoin d'un serveur XMPP pour marcher. Google recommande Openfire mais publie aussi une documentation <http://code.google.com/p/wave-protocol/wiki/Installation> pour le faire marcher avec ejabberd. (Et des gens y sont arrivés avec Prosody <http://code-bear.com/bearlog/2009/11/01/google-wave-fedone-component-with-prosody-xmpp-server/>).

Les serveurs de googlewave.com n'acceptaient d'ailleurs pas de parler avec d'autres, au lancement du service (cela a changé le 2 novembre <http://googlewavedev.blogspot.com/2009/11/wavesandboxcom-federate-this.html>, avec la vraie ouverture de la fédération).

S'il n'existe pas de protocole standard pour le client, il y a par contre une API pour écrire des robots qui parleront à Google Wave et un protocole sous-jacent (RPC avec JSON). Il y a par exemple du code pour un robot <http://code.google.com/p/wave-robot-python-client/> en Python mais qui ne marche apparemment qu'avec l'App Engine. Des robots plus ou moins utiles apparaissent déjà, comme celui qui fait la passerelle avec Twitter <http://www.webupd8.org/2009/11/how-to-post-to-twitter-from-google-wave.html>.

Pour résumer, une technique très prometteuse, qui avait de bonnes chances de succès, qui est largement ouverte, mais qui était très loin d'être finalisée et qui ne le sera probablement pas <http://googleblog.blogspot.com/2010/08/update-on-google-wave.html>. Le code a été publié en logiciel libre, quelques développements <http://googlewavedev.blogspot.com/2010/09/wave-open-source-next-steps-wave-in-box.html> sont encore prévus. La maison officielle de Google Wave est désormais l'incubateur Apache <http://www.h-online.com/open/news/item/Google-Wave-accepted-into-Apache-Incubator-1147935.html>.

Parmi les articles intéressants :

- « Pourquoi Wave est intéressant + un guide » <http://pisani.blog.lemonde.fr/2009/11/25/pourquoi-wave-est-interessant-un-guide/> »
- « *Google Wave, or wikis, could be used to build a software forge* » http://www.jroller.com/dmdevito/entry/google_wave_or_wikis_could »

- « Utiliser Google Wave en classe, premier bilan <<http://www.histobric.fr/Utiliser-Google-Wave-en-c>> une expérience pédagogique avec Google Wave
- Il existe un tutoriel Google Wave payant (six dollars états-uniens), « *The Complete Guide to Google Wave* » <<http://completewaveguide.com/>> » de Gina Trapani, que j'ai acheté via Google Checkout. Il est plutôt bien fait mais il parle de Google Wave (le client Google) pas de Wave : zéro contenu sur la fédération, ou sur les autres serveurs et clients. C'est surtout un mode d'emploi du client Google Wave, avec beaucoup de copies d'écrans.