

# Comet

Depuis quelques semaines, je cherche un moyen permettant à mes sites de proposer des interactions en temps réel avec les utilisateurs. Il y a bien sûr toutes les techniques liées à AJAX, mais elles sont gourmandes en nombres de requêtes lorsqu'on veut, par exemple, gérer un chat. D'ailleurs en ce qui concerne les chats, on se trouve rapidement confronté au problème de latence du réseau, ce qui fait que dans la plupart des applications testées, les messages arrivent avec un certains délais. On pourrait utiliser Flash pour ce genre d'application, mais il faut un plugin et généralement les sites basés sur cette technologie sont lourds.

En cherchant sur le web, on trouve un autre ensemble de « méthodes » permettant de gérer des communications bi-directionnelles (serveur-client et client-serveur) : il s'agit de Comet.

J'ai donc rapidement étudié cette

technique (vous trouverez quelques infos sur Comet Daily), ce qui m'a amené à installer le serveur Meteor. Le choix de ce serveur n'est pas un hasard : cette page est un bref comparatif des solutions actuelles et j'en ai déduit que Meteor était le plus approprié pour mon utilisation.

Après quelques tests, je me suis rendu compte de ses limitations (ainsi que du support quasi-nul) et j'ai donc passé mon chemin. J'ai trouvé une autre solution, appelée APE, pour Ajax Push Engine. Il s'agit d'un serveur codé en C, récent et qui supporte des extensions en JavaScript du côté serveur. Un framework client est mis à disposition et permet de communiquer avec le serveur assez facilement.

Des performances flatteuses (il n'y a pas de benchmark « officiel », mais c'est ce qui ressort dans les différents articles que j'ai vu sur le web), peu de limitations et beaucoup d'avantages :

- modularité du serveur,

- la possibilité de créer des applications cross-domains,
- le choix du mode de transport (JSONP, XHRStreaming, XMLHttpRequest etc...),
- un développement actif

Prochain billet : l'installation du serveur APE.