DALLA-NORA Enzo
Greaux Thomas

Miniprojet Java RMI

Le protocole HTTPStreaming garde une connexion ouverte durant toute la durée du stream. Ce qui se traduirait en Java RMI par l'exécution d'une unique méthode. Ce qui implique que soit le serveur ne peut gérer qu'un client à la fois, soit mettre en place du multithreading ce qui n'est pas satisfaisant dès qu'il y a un nombre conséquent de clients.

Le principal problème avec RMI est qu'il fonctionne sur un système de callback, et que nous n'avons pas la main mise sur ce mécanisme, notamment à cause du niveau d'abstraction de la technologie. De ce fait, l'implémentation d'un web hook qui repose sur ces callbacks et ces fermetures de connexion sont particulièrement adapté pour RMI. A contrario, le http streaming demande de garder cette connexion pendant la durée du transfert, ce qui est très compliqué à faire/simuler.

Il faudrait, pour l'implémenter, réussir à passer outre la limitation RMI en ayant un client et un serveur qui échangent en continue, mais il resterait toujours ce problème de connexion interrompue. La principale barrière pourrait être « effacée » avec l'implémentation d'une boucle d'appel serveur du coté client, qui permettrait de contourner l'interruption de connexion et la non utilisation des callbacks. Cependant, cette implémentation possède aussi ses limites, puisqu'elle ne retransmet pas exactement ce mécanisme de http streaming puisque le client doit appeler le serveur a intervalle régulier pour relancer la connexion et donc la réception des différentes données.