Outil de modélisation de performances des migrations de données inter-cloud

Antoine Martin - Carole Bonfré

Février 2015

Résumé : Ce projet a pour but de permettre de choisir, selon une localisation géographique et des paramètres donnés, la meilleure solution d'hébergement *Cloud* existante. De nombreux chercheurs rencontrent le besoin de déplacer des Téraoctets de données, cet outil leur permettra donc d'optimiser la migration de leurs données. À terme, ce projet Open Source proposera une solution unique sur le marché.

I Introduction

Dans un monde toujours plus interconnecté, nos données sont de plus en plus dématérialisées et éparpillées. Nous sommes amenés à les transférer d'un hébergeur à un autre et dans le but d'optimiser ces opérations, il nous a été demandé de nous poser différentes questions concernant l'évaluation des méthodes de transfert.

Celas'inscrit dans le cadre de l'équipe de recherche Avalon du LIP de l'ENS de Lyon, qui propose des solutions pour la distribution des calculs dans des fédérations de *Cloud*. La distribution de ces calculs implique de nombreux mouvements de données inter-cloud. Pour contribuer à l'amélioration des travaux de recherche des membres de l'équipe Avalon, nousallonsdévelopper un outil permettant d'évaluer la performance de transferts de données entre plusieurs hébergeurs de "Cloud". Cet outil permettra soit d'effectuer des tests de performances « à la demande », soit de récupérer les résultats déjà obtenus lors de précédents tests.

Ce projet, nous l'espérons, pourra non seulement aider les chercheurs de l'équipe Avalon de l'ENS mais également offrir un outil à tous projets ou personnes ayant besoin d'évaluer les temps de transfert entre solutions *Cloud* (académiques comme privées).

II Recherche et Analyse

Une première étude des solutions existantes nous aura permis de sélectionner les outils les plus cohérents pour définir la structure des données de notre application.

A Étude de l'existant

Cette étude peut être divisée en deux parties. Tout d'abords, nous avons détaillé les offres de solution de stockage "Cloud" disponibles sur le marché afin de pouvoir en sélectionner. Nous avons ensuite cherché un outil déjà capable de réaliser des tests de performances sur des "Clouds"pour justifierle développement de notre propre outil.

YC : Cela quoi ? La thématiqu le travail, stage ?

YC : étudier, p poser, et développe

YC : pourquoi futur ? C' maintenar du passé. Donc soit nous avon été amené à... soit da ce travail, nous étu- dions+pro un outil q

YC : Ça f fumeux et n'apporte pas d'info mation su la suite. S pas utile, sucre. Sin mettez de l'informa- tion sur c

1 Solutions de stockage *Clouds*

TABLE 1 – Tableau comparatif des "*Clouds*"

drive	API	Emplacement	Libre	Espace de stockage	Limitation	SDK
Dropbox	Oui	S3	Gratuit / Propriétaire	2Go	Oui(N/A)	Oui
Google drive	Oui	lien	Gratuit / Propriétaire	15Go	10 000 requêtes /jour, 10 requêtes /sec/user	Oui
S3	Oui	lien	Gratuit / Propriétaire	5Go	20 000 GET, 2 000 PUT / mois	Oui
Onedrive	Oui	?	Gratuit / Propriétaire	15Go	Oui (NA)	Oui
Cloud Orange	Oui	Paris (Sénégal ?)	Gratuit / Propriétaire	10Go à 100Go	Oui (2Go par fichier)	Oui
Hubic	Oui	France (Paris, Roubaix)	Gratuit / Propriétaire	25Go	Oui (10Go par fichier)	Oui
Microsoft Azure	Oui	lien	Gratuit / Propriétaire	100To	Oui (NA)	Oui
iCloud	Oui	USA (Caroline du Nord)	Gratuit / Propriétaire	5Go	Oui (15Go par fichier)	Oui
Google Cloud Storage	Oui	lien	Payant / Propriétaire	1To	Oui 5Tb par fichier	Oui
Cloud bouygues	Non	USA (Pogoplug)	Payant / Propriétaire	5Gb	Oui (NA)	Non
SFR Cloud	Non	Paris	Payant / Propriétaire	100Go	Oui (NA)	Non

*La mention Oui(N/A) pour la colonne des limitations signifie une présence de limitation non explicitée par l'hébergeur.

Nous avons remarqué lors de notre analyse que l'hébergeur Dropbox utilise en fait les services d'Amazon S3. Dropbox étant un des services les plus populaires, nous avons décidé de le choisir tout demême car il apporte probablement des services supplémentaires. Nous pensons qu'il est également intéressant de tester les différences de performance entre Amazon S3 et Dropbox. Pour les mêmes raisons, il aurait également été intéressant de pouvoir vérifier les différences entre Google Drive et Google Cloud Storage.

YC : perso
tout de
même – et
+ ' :'

Nous avons aussi étudié la possibilité de sélectionner OwnCloud parmi les hébergeurs (mais l'utilisateur doit posséder une machine avec OwnCloud installé). Il ne possède pas de limitations particulières, l'espace de stockage est "infini" (il dépend du serveur), et nous avons trouvé un SDK développé par un tiers qui semble exploitable pour notre application. À terme, l'ajout de OwnCloud peut donc être envisagé. Après analyse des différents acteurs du marché de stockage en ligne, nous avons décidé de sélectionner les trois hébergeurs suivant : Dropbox, Amazon S3 et Google Drive. Ils possèdent tous les trois des SDK qui facilitent le développement de notre outil. En revanche, les espaces de stockage sont parfois assez limités. Nous souhaiterions, à terme, ajouter Google Cloud Storage pour les raisons citées précédemment.

YC : comm
il doit avo
un compte
drive.
Présentez
Owncloud
comme un
solution li
bre pour
mettre en
place un
cloud priv
Owncloud

2 Solutions d'évaluation de performance des stockages *Cloud*

Trois projets ont été identifiés durant cette étude. Le premier, HP Performance, est une solution propriétaire payante. Parmi de nombreux outils, elle propose de se placer dans une zone géographique pour provisionner un générateur de charges (il est donc possible de chercher le meilleur emplacement). Le second projet, COSBench, est une solution libre mais plutôt limitée puisqu'elle ne concerne que Swift Storage et Amazon S3. Elle permet d'exécuter des tâches sur des outils distants et de les surveiller. Enfin, CloudScreener est une solution payante que nous n'avons pas pu tester (nous ne connaissons donc pas la précision de cet outil). Nous les avons comparés à notre projet baptisé Kyd (Know Your Data).

Table 2 – Tableau comparatif des solutions trouvées

	HP Performance	**CosBench**	**CloudScreener**	**Kyd**
Générique	Oui	Oui	Non	Oui
Open source	Non	Oui	Non	Oui
Modulaire	Non	Non	Probablement	Oui
Interface graphique	Oui	Oui	Oui	Non
Limites	Propriétaire	Swift Storage et S3	Propriétaire, tests spécifiques	limites des *Clouds*
Stockage des résultats	Exports multiples	Exports multiples	Web	Base de données

Un autre outil en cours d'implémentation a aussi attiré notre attention : PerfKit. Développé par Google, il est très similaire à notre projet mais comporte des différences majeures puisqu'il effectue ses tests seulement pour des machines virtuelles. Il nécessite de pouvoir installer des logiciels sur les serveurs des *Clouds*, ce qui n'est pas réalisable puisqu'il faut avoir l'accord des hébergeurs (le projet se limite donc aux propres hébergeurs de Google et, actuellement, à Microsoft Azure et Amazon AWS qui sont des serveurs virtuels privés). Ce projet ne peut pas atteindre les hébergeurs que nous ciblons et ne constitue donc pas un concurrent à proprement parler. On peut donc remarquer que l'application que nous allons développer se situe sur un secteur qui n'est pas encore exploité, ce qui veut dire que nous proposerons une solution unique.

B Définition de la structure et modélisation

Un grand nombre de paramètre est à prendre en compte pour obtenir des résultats cohérents et réutilisables. Il faut donc connaître la taille d'un fichier transféré, l'emplacement géographique de l'utilisateur, les protocoles utilisés, la date du test, les hébergeurs à tester, ainsi que le type de transfert (*upload* ou *download*) pour sauvegarder les résultats avec précision. En retour, l'application propose une liste des différents hébergeurs testés avec les temps obtenus, le meilleur choix étant mis en valeur. Tous les résultats calculés sont stockés en base de données pour pouvoir être réutilisés lors de tests ultérieurs.

C Choix des technologies

Le fait de travailler avec beaucoup de paramètres différents impose de pouvoir tester toutes les solutions possibles de manière exhaustive. Il fallait également pouvoir stocker ces dernières de la façon la plus adapté possible pour pouvoir les retrouver facilement. Deux outils ont donc été retenus : Execo Engine et MongoDB.

1 Execo engine

Execo est un outil développé par les membres de l'ENS ayant de multiples fonctionnalités. Dans notre cas,

YC : pas clair pour le lecteur lambda

YC : Quel est le rapport avec ce que vous faîtes ? Pr de perf ?

YC : mais vous pouvez décrire un peu ses fonctionnalités ? Que le lecteur comprenn pourquoi vous le cit ici, en quo votre outil répond à trucs qu'e ne font pê pas

YC : ça serait bien de parler début que le lecteur peut trouv un résumé des caractéristiques de projets et d'y fair référence ! (avec les command ref et labe

YC : plutôt que de par ler de con current, la date de m en chantie ou quand c'est deve public par rapport au début de votre trav

YC : une

seule la partie Enginede Execo est intéressante puisqu'elle permet de combiner tous les paramètres passés par l'utilisateur pour tester les différentes possibilités de test. Par exemple, si l'utilisateur souhaite tester le transfert de fichiers de différentes tailles sur les différents hébergeurs, Execo Engine génère les combinaisons "taille1/drive1", "taille1/drive2", *etc.* Toute combinaison n'ayant pas pu être testée à cause d'une erreur est enregistrée et peut être relancée plus tard. Il s'agit donc d'un excellent outil de test qui chronomètre également tous les tests effectués de façon très précise.

YC : parle de produit cartésiens pour la génération de l'ensem ble des co figurations possibles ?

2 MongoDB

Notre application possédant une structure de données appelée à être modifiée régulièrement, il fallait qu'elle possède un système de gestion de base de données adapté. Sachant qu'une seule table serait nécessaire mais qu'elle contiendrait, à terme, un très grand nombre de tuples, nous nous sommes tournés vers le système non relationnel qu'est MongoDB. Sa vitesse de traitement associée aux index permet de requêter très rapidement pour fournir un résultat à l'utilisateur.

III Implémentation

Kyd est une application implémentée en langage Python. Son développement se divise en deux parties : l'interaction avec les hébergeurs et l'interaction avec l'utilisateur. Pour plus d'efficacité, nous avons implémentéensemble toutes les parties sur Dropbox puis nous nous sommes partagés le travail sur Amazon S3 et Google Drive.

YC : ou le nom de l'outil et son rôle en titre, serai meilleur

A Interaction "*Clouds*"

Dans un premier temps, il nous a fallu mettre en place toutes les connexions avec les hébergeurs sélectionnés. Nous avons utilisé les SDK propres à chaque *Cloud* et nous avons ensuite effectué une série de tests unitaires pour vérifier le fonctionnement du transfert de fichiers en *download* et en *upload*.

YC : donc 2 APIs ? S oui, le dir car leur d sign doit être traité

Cette partie du code se veut très modulaire puisqu'elle doit permettre l'ajout d'autres hébergeurs. Pour faciliter ces ajouts, nous avons créé une classe par hébergeur (il suffit donc d'implémenter les fonctions désirée pour effectuer un ajout). Nous avons rencontré quelques difficultés durant cette étape puisqu'une telle implémentation demande de s'adapter à l'utilisation de chaque SDK. Malgré tout, les hébergeurs sélectionnés étant très populaires, ils disposent d'une large communauté qui permet de régler rapidement la majorité des problèmes.

YC : mot qui revien trop

B Interaction utilisateurs

Actuellement, l'interface de communication avec l'utilisateur se fait sous console. L'utilisateur demande à l'application la meilleure solution de transfert de données selon ses paramètres et, si la base de données MongoDB contient des informations suffisamment similaires, le résultat est retourné à l'utilisateur. Dans le cas contraire, Kyd demande à l'utilisateur s'il accepte d'effectuer un test selon certains paramètres donnés pour enrichir la base de données et pouvoir répondre lors d'une prochaine requête. Pour minimiser le nombre de paramètres à saisir pour l'utilisateur, sa localisation géographique est automatiquement détectée grâce à son adresse IP et à deux API de géolocalisation (ip-api et Telize qui se relaient si le serveur de l'une d'entre elles est hors service).

YC : sous console ?

YC : suff- isamment vague

Le schéma ci-dessous synthétise le fonctionnement de notre application dont les divers éléments ont été expliqués précédemment.

YC : décrivez e 2 phrases un scénar pour savoi qui fait qu et quand, l'ordre des actions, comme à quel mom execo est utilisé car jamais ex- plicitemen dit.

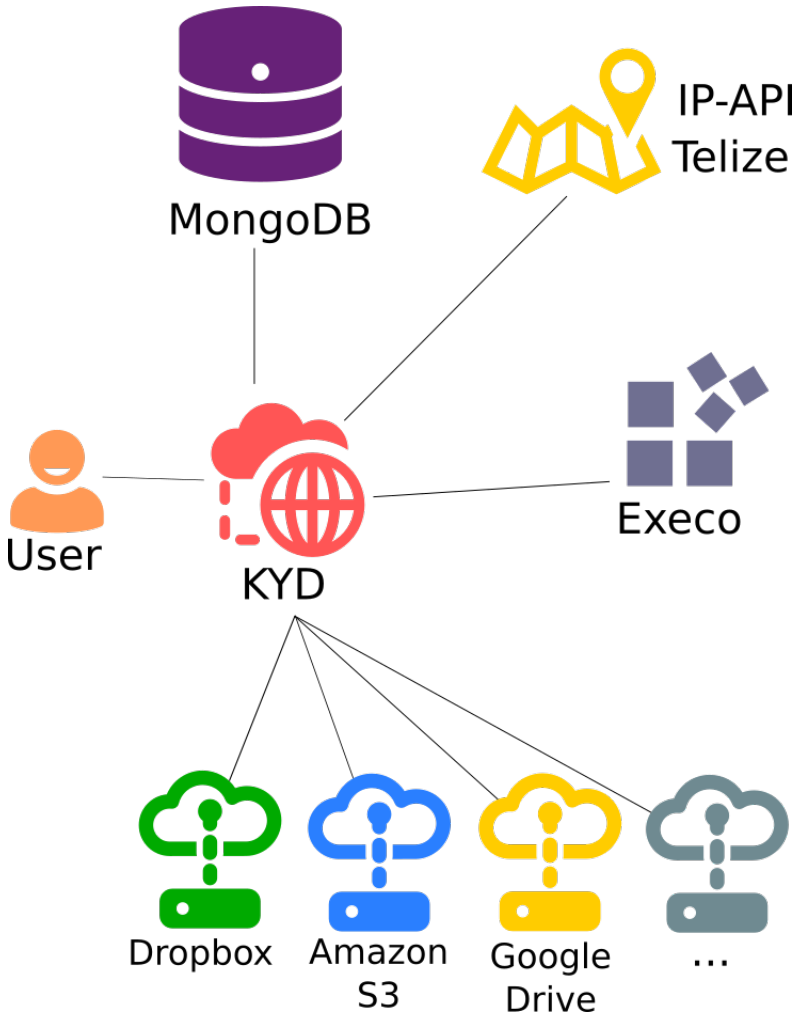

FIGURE 1 – Architecture de l'application

C Améliorations

À terme, plusieurs améliorations sont envisageables. Ajouter des paramètres pour augmenter la précision des tests ferait partie de ces améliorations. Il serait par exemple possible de vérifier l'efficacité des compte premium qui apportent peut être de meilleures performances. Il faudrait également compléter la base de données pour pouvoir répondre le plus souvent possible à l'utilisateur sans avoir à lui faire effectuer des tests parfois coûteux en temps. Il serait alors nécessaire d'effectuer des tests à plusieurs emplacements géographiques dans le monde.

Enfin, une interface Web plus conviviale pourrait rapporter les résultats à l'utilisateur à la place de la console. Cette amélioration se veut majoritairement esthétique mais pourrait apporter un certain nombre de renseignements supplémentaires à l'utilisateur.

IV Analyse des résultats

Suite à de nombreux tests, nous avons pu obtenir différentes courbes traduisant les divers aspects de notre projet. Nous désirions comparer les trois hébergeurs pour une localisation donnée, mais nous voulions aussi savoir s'il existait des différences selon le moment de la journée ou selon le jour de la semaine.

Pour les figures 1 et 2, les tests ont été réalisés depuis le campus de la Doua à Lyon 1. Un grand nombre de tests ont été effectués en *upload* et en *download* avec une taille de fichier différente puis la moyennede ces tests a

YC : pourquoi, actuelleme pas assez précis ? C une fausse question, mais justi ce que vou écrivez

YC : parle d'ergonom de synthès d'informa- tion

YC : décrivez calmemen le protoco expérimen tal. Quels

permis d'obtenir ces résultats.

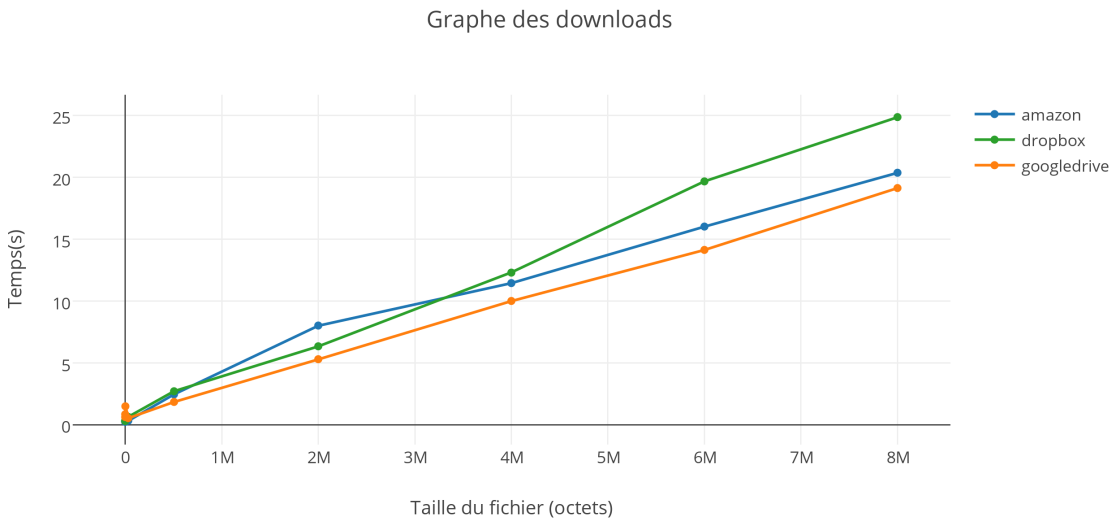

FIGURE 2 – Graphe des *downloads*

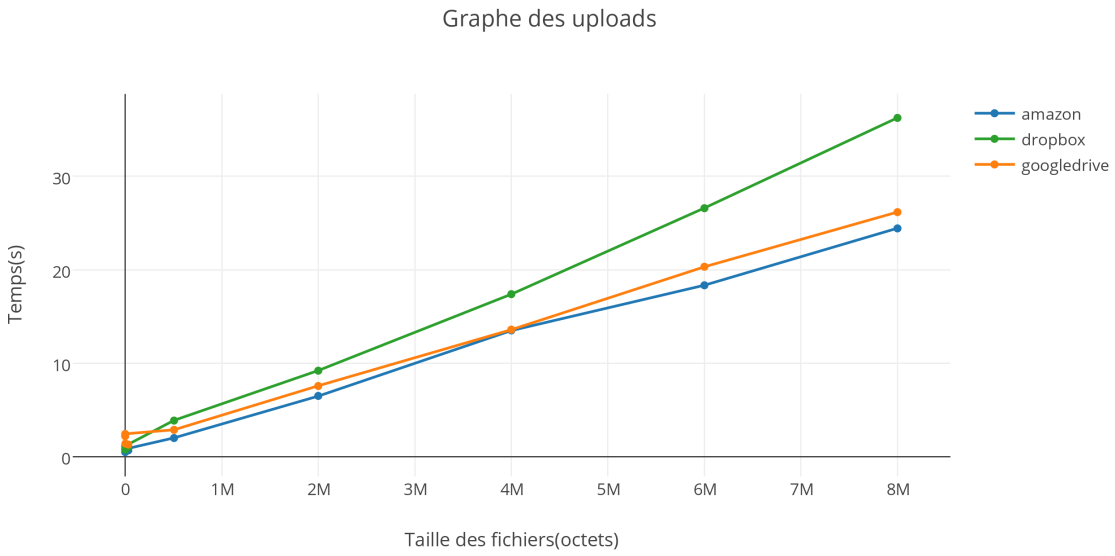

FIGURE 3 – Graphe des *uploads*

Il est clairement visibleque, dans cette situation, Dropbox est le choix le moins pertinent sauf en ce qui concerne la connexion. Amazon et Google Drive seront donc des choix beaucoup plus logiques puisqu'on peut observer une différence d'environ 20 pointsentre Dropbox et ses deux concurrents en ce qui concerne les performances (et cette différence ne cesse de croître avec l'augmentation de la taille des fichiers).

Pour les deux figures suivantes, les tests ont été effectués depuis un serveur virtuel situé à Londres. Ils ont été lancés toutes les heures pendant vingt quatre heures un dimanche avec une taille de fichier de 10Mo à l'aide de la crontab du serveur.

YC : pê. Mais après que vous ayez présenté l courbes, et où le lecteur vo clairement Et avec référence au(x) tableaux/ qui vont bien

YC : des points ?

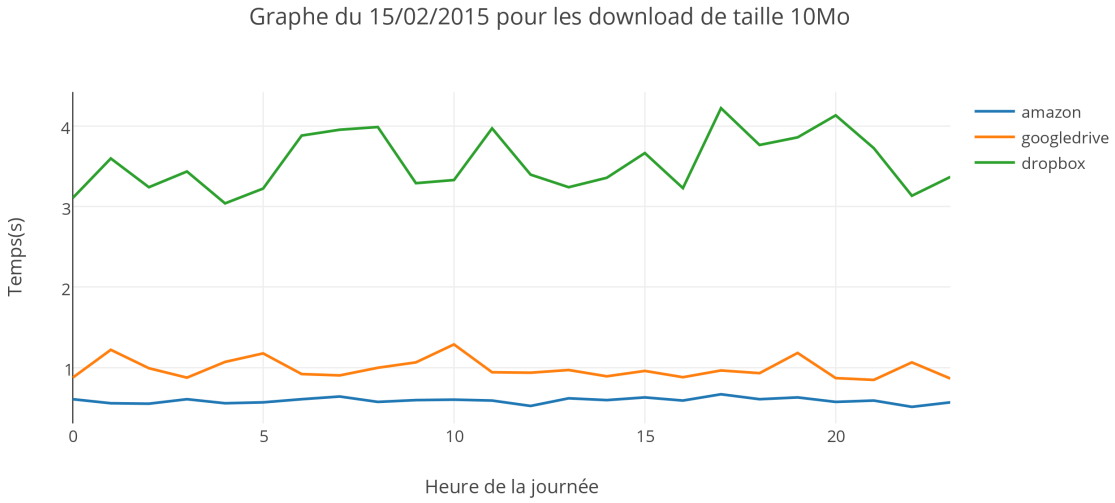

FIGURE 4 – Graphe des *downloads* du 15 de taille 10Mo

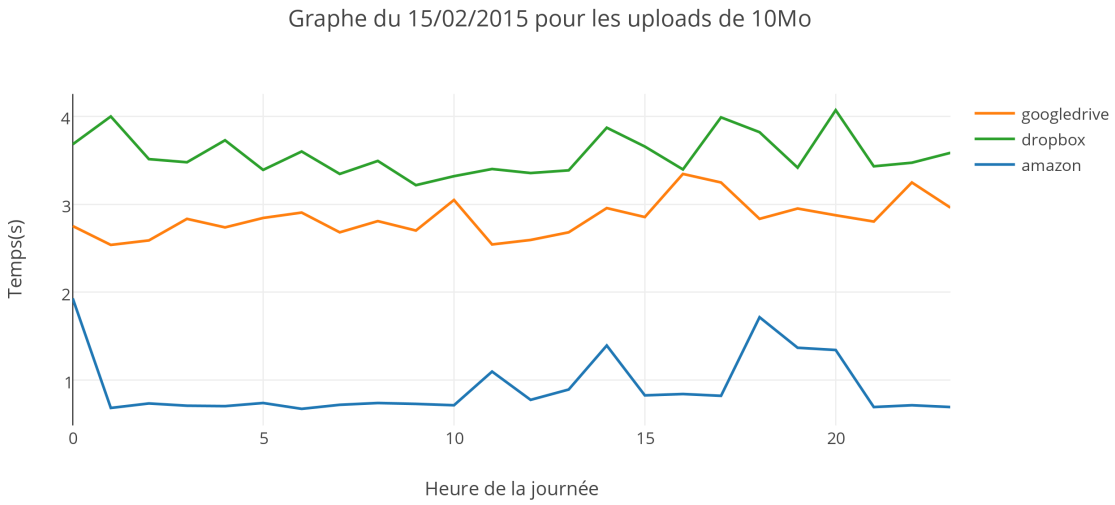

FIGURE 5 – Graphe des *uploads* du 15 de taille 10Mo

En *upload*, on observe qu'Amazon suit un schéma connu qui comporte des pics de latence à 11h, 14h puis entre 18h et 20h. On suppose que cette situation est due au grand nombre de connexions simultanées durant une plage horaire plus bondée que les autres. Les performances de Dropbox sont, cette fois encore, très nettement inférieures à celles de ses concurrents.

YC : du bon commentaire. J'en veux d'autres !

V Conclusion

Durant ce projet, nous avons donc implémentéun outil permettant d'estimer la meilleure solution de transfert de données entre plusieurs hébergeurs. Pour ce faire, une phase de recherche était nécessaire puisqu'il fallait vérifier qu'un tel outil n'existait pas déjà et, dans ce cas, sélectionner les meilleures solutions "*Cloud*".

YC : ouch CE qu'il ne faut pa faire...

Les résultats obtenus suite aux tests ont permis de vérifier la validité de notre application. En effet, un écart trop important entre les résultats aurait entraîné un manque d'homogénéité des moyennes. Dans une telle situation, KYD n'aurait pas pu fournir de résultat valable et nos recherches auraient prouvées qu'un tel outil n'était pas réalisable.

YC : replacez la problématique, votre étud puis votre développe ment. Et ce que le développe ment à permis de mesurer

VI Annexes

Ce que vous avez vu, ce sur quoi vous vous êtes formés, des choses que vous avez apprises (travail d'initiation à la recherche, l'évaluation devrait aussi comporter un paragraphe en annexe sur vos impressions, votre retour d'expérience).

YC : vous parlez rep ductibilité Pas top cl

YC : dépe de votre réponse avant, ma à priori, à retirer