



**HAL**  
open science

## Towards an architecture for tag-based predictive placement in distributed storage systems

Stéphane Delbruel

► **To cite this version:**

Stéphane Delbruel. Towards an architecture for tag-based predictive placement in distributed storage systems. Web. Université de Rennes, 2017. English. NNT : 2017REN1S006 . tel-01523568

**HAL Id: tel-01523568**

**<https://theses.hal.science/tel-01523568>**

Submitted on 16 May 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**THÈSE / UNIVERSITÉ DE RENNES 1**  
*sous le sceau de l'Université Bretagne Loire*

pour le grade de  
**DOCTEUR DE L'UNIVERSITÉ DE RENNES 1**

*Mention : Informatique*  
**Ecole doctorale Matisse**

présentée par

**Stéphane Delbruel**

préparée à l'unité de recherche UMR 6074 IRISA  
Institut de recherche en informatique et systèmes aléatoires  
ISTIC/IRISA

---

**Towards an  
Architecture  
for Tag-based  
Predictive  
Placement in  
Distributed  
Storage  
Systems**

**Soutenance prévue le 27 janvier 2017**  
devant le jury composé de :

**Danny HUGHES**

Professeur, KU Leuven / rapporteur

**Romain ROUVOY**

Professeur, Université de Lille / rapporteur

**Sonia BEN MOKHTAR**

Chargée de Recherche, CNRS / examinateur

**Antonio CARZANIGA**

Professeur, USI Lugano / examinateur

**Davide FREY**

Chargé de Recherche, Inria / examinateur

**Guillaume PIERRE**

Professeur, U. de Rennes 1 / examinateur

**François TAÏANI**

Professeur, U. de Rennes 1 / directeur de thèse



# Résumé en Français

## Introduction

Cette thèse rassemble et présente la part la plus importante de mes recherches durant ces dernières années, portant sur les systèmes de stockage décentralisés. Dans cette thèse, nous affirmons que les méta-données générées par les utilisateurs dans les systèmes pleinement distribués de gestion de contenu générés par les utilisateurs représentent une source d'informations fiable nous permettant de déterminer lors de la mise en ligne d'un contenu, où il sera consommé dans le futur. Cela rend donc possible le placement prédictif de contenus proche de ses futurs consommateurs, augmentant par là-même la pertinence des caches de proximité.

**Motivation** Internet a fondamentalement modifié la manière dont les humains communiquent et ces changements introduits par ce nouveau médium sont fréquemment comparés à ceux amenés par l'apparition du verbe, de l'écriture ou de l'imprimerie [52]. Tels ces médias auparavant, Internet a modifié l'étendue et la portée des possibilités de communication précédemment offertes, et a défini l'émergence d'un âge où l'information connaît un renouveau de croissance et de diffusion. Ce médium en n'étant pas contraint par des frontières géographiques, et en aplanissant à importance égale la capacité de tout un chacun à s'exprimer et à être entendu après des décennies de domination par les médias traditionnels et les gouvernements, a été considéré comme une menace possible à leur propre souveraineté [51].

Plusieurs tentatives de censure et de contrôle ont été tentées depuis ses débuts, que ce soit pour forcer un contrôle étatique sur chaque contenu publié <sup>1</sup>, pour

---

<sup>1</sup><http://www.conseil-constitutionnel.fr/conseil-constitutionnel/francais/les-decisions/acces-par-date/decisions-depuis-1959/1996/96-378-dc/decision-n-96-378-dc-du-23-juillet-1996.10818.html>

empêcher des citoyens de communiquer avec le reste du monde durant des événements populaires [19] ou bien pour exercer une surveillance massive et intrusive piétinant l'Article 8 de la Convention Européenne des Droits de l'Homme [63]. Toutefois, une menace supplémentaire provenant d'acteurs non-étatiques grandit, et représente à mon sens un plus grand danger à long terme. Il s'agit de la centralisation et du contrôle des données personnelles des utilisateurs, par des acteurs privés. La collecte de masse des données a permis de construire de grands data sets, et leur exploitation dans des domaines aussi variés que la recommandation de contenu, les modèles prédictifs et récemment l'apprentissage des intelligences artificielles. Une des principales nécessités face à cette situation est donc de développer des modèles alternatifs pour pallier aux services offerts par ces grands acteurs privés, tout en s'assurant que ces services alternatifs soient distribués et respectueux de la vie privée de leurs utilisateurs. Ma thèse s'inscrit dans cette dynamique et vise à proposer dans le cadre des grandes plateformes de distribution de contenu généré par des utilisateurs tiers, une solution alternative et distribuée où aucun acteur n'amasse une connaissance globale des actions du système, en se concentrant sur le placement prédictif des contenus au plus près des utilisateurs.

**Contexte** Les travaux présentés dans cette thèse touchent en particulier trois domaines distincts. Le premier plutôt généraliste est celui des systèmes de stockage distribués. A partir de la deuxième génération de ces systèmes, post-Gnutella et Napster, on note l'introduction de caractéristiques intelligentes telles qu'un fort accent sur la vie privée pour Freenet [21] ou bien sur la gestion du stockage et des caches de distribution dans PAST [59]. Enfin, récemment, Storj [64] va au-delà et réussit à réunir des composants clés des deux systèmes précédents tout en les mariant avec des technologies émergentes comme une chaîne de bloc, permettant d'aller toujours plus loin dans les systèmes décentralisés de stockage de masse.

Le deuxième domaine abordé est celui de la distribution de contenu, historiquement abondamment commenté entre toutes les études de différentes architectures pour les réseaux de distribution. Le troisième domaine met l'accent sur le rôle des méta-données, à travers les schémas de consommation et les propriétés géographiques exploitables.

## Contributions

Dans cette thèse, nous mettons en avant l'exploitation des méta-données attachées volontairement par les producteurs de contenus, avec une attention particulière portée sur les tags. L'objectif est de démontrer les bénéfices et la viabilité d'un système distribué comportant un mécanisme de placement prédictif. Pour cela, trois contributions incrémentielles sont présentées et résumées ci-dessous.

**Analyse de données.** Une analyse extensive est faite sur un grand jeu de données de YouTube. Elle met en avant une relation claire entre la distribution géographique de certains tags et la distribution géographique des vues des vidéos qui leurs sont relatées. De plus il apparait qu'une grande proportion des vidéos présentes dans ce jeu de données ainsi que leurs tags associés ont une distribution géographique très étroite de leurs vues. Une analyse entropique suggère que les contenus ayant une entropie faible sont associés à des marqueurs ayant également une faible entropie. Cette analyse et les diverses observations qui en découlent, pointent vers la possibilité de se baser sur ces marqueurs pour identifier où et dans quelles proportions un contenu sera consommé à travers le monde.

**Le potentiel prédictif des tags.** Cette deuxième contribution introduit plusieurs approches visant à déterminer si il est possible et efficace de s'appuyer sur les marqueurs précédemment énoncés comme leviers de prédiction de consommation, ainsi qu'une évaluation de l'efficacité d'un tel système de placement proactif. Les approches proposées sont volontairement simples et peu coûteuses, permettant d'obtenir des résultats probants tout en laissant de la place à de futures optimisations. La deuxième partie de cette contribution démontre que de telles prédictions peuvent aider à placer du contenu fraîchement produit dans un système distribué de distribution de contenu, le tout en utilisant des ressources limitées et une infrastructure globalement distribuée. Une des limitations de cette contribution est son besoin de reposer sur une centralisation de la connaissance de la consommation des contenus pour être accessible par chaque acteur. Cette limitation est levée dans la troisième contribution.

**Protocole d'estimation décentralisée.** Dans cette dernière contribution, nous introduisons Mignon, un nouveau protocole ainsi qu'une architecture qui permettent une estimation rapide d'un agrégat d'affinités envers un contenu fraîche-

ment produit au sein d'une communauté d'utilisateurs, ainsi que de placer de manière pleinement décentralisée le contenu en conséquence. Notre proposition évite une agrégation coûteuse et explicite en utilisant les propriétés des réseaux superposés auto-organisés basés sur des similarités, et peut être utilisée au sein des systèmes pleinement distribués de gestion de contenu généré par les utilisateurs. Le noyau de ce protocole réside dans la capacité à estimer l'intérêt des nœuds présents au sein d'une zone géographique délimitée, en se basant seulement sur des bribes d'informations obtenues via les réseaux superposés auto-organisés. Parmi les diverses solutions étudiées, l'utilisation de la méthode d'interpolation de Gregory-Newton a retenu notre intérêt. Notre proposition d'architecture est désormais libre de tout point central, et permet à n'importe quel acteur du système de déterminer de manière rapide, peu coûteuse et efficace où dans le monde son contenu produit sera consommé dans le futur.

## **Conclusion**

Cette thèse propose une solution à certains problèmes ouverts relatifs aux systèmes décentralisés de stockage de contenu. Les trois contributions forment un tout cohérent proposant une nouvelle approche pour des systèmes de placement pro-actif de contenu proche des utilisateurs, dans une optique de diminuer les coûts associés. Cette solution se veut prendre place au sein de systèmes alternatifs ouverts, visant à participer à l'amélioration globale de leurs performances, dans leur compétitions contre les systèmes centralisés et propriétaires. Cette solution s'adresse à des systèmes distribués avec une granularité de l'ordre de l'ordinateur personnel ou bien du boîtier décodeur. La piste sur laquelle repose cette solution non conventionnelle n'a à notre connaissance à ce jour pas déjà été exploitée. Nous croyons fermement qu'un mécanisme de placement prédictif au sein d'un système distribué de gestion de contenu généré par ses utilisateurs, qui place les contenus produits au sein des zones géographiques où ce contenu a le plus de chance d'être consommé aide en maximisant la part locale de l'ensemble du trafic et permet donc de rendre de façon générale le réseau plus flexible et donc plus apte à accomplir d'autres tâches.

# Contents

<b>Résumé en Français</b>	<b>i</b>
Introduction . . . . .	i
Contributions . . . . .	iii
Conclusion . . . . .	iv
<b>Contents</b>	<b>vii</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xi</b>
<b>I Context</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Motivation . . . . .	4
1.1.1 From assessment to research . . . . .	4
1.1.2 The rise of User-Generated Content . . . . .	7
1.2 Problem statement . . . . .	9
1.2.1 Tags as pertinent metadata . . . . .	11
1.2.2 Tags as reliable indicator for predicting consumption patterns	12
1.2.3 Decentralized predictive placement system . . . . .	13
1.3 Contributions . . . . .	14
1.3.1 Metadata analysis of a UGC service . . . . .	15
1.3.2 Predictive power of tags . . . . .	15
1.3.3 Distributed prediction mechanism . . . . .	15
1.4 List of Publications . . . . .	16
1.5 Organization of the document . . . . .	17
<b>2 Related Work</b>	<b>19</b>



2.1	Distributed storage systems . . . . .	20
2.1.1	Freenet . . . . .	20
2.1.2	PAST . . . . .	22
2.1.3	Storj . . . . .	23
2.2	Content distribution . . . . .	24
2.2.1	Hybrid CDNs . . . . .	24
2.2.2	Content Delivery for alternate systems . . . . .	25
2.3	Metadata in UGC services . . . . .	25
2.3.1	Consumption patterns . . . . .	26
2.3.2	Geographic properties . . . . .	28
2.4	Summary . . . . .	29
<b>II Thesis Contributions</b>		<b>31</b>
<b>3</b>	<b>Dataset Analysis</b>	<b>33</b>
3.1	Presentation and preparation of the dataset . . . . .	34
3.1.1	Notation . . . . .	35
3.1.2	From popularity to number of views . . . . .	36
3.1.3	Metrics used to analyze the dataset . . . . .	37
3.2	Tag and view distributions . . . . .	38
3.3	Comparing the Entropy of Videos and Tags . . . . .	41
3.3.1	Video popularity and geographic distribution . . . . .	41
3.3.2	Tags and geographic distribution . . . . .	43
3.3.3	Tags as reliable markers of geographic specificity . . . . .	45
3.4	Possible mitigation of the results . . . . .	47
3.5	Summary . . . . .	48
<b>4</b>	<b>Potential of the tags</b>	<b>49</b>
4.1	Predicting views shares from tags . . . . .	50
4.1.1	General approach . . . . .	50
4.1.2	Baseline . . . . .	51
4.1.3	Evaluation and metric . . . . .	51
4.1.4	Results . . . . .	53
4.2	Potential of tags for proactive video placement . . . . .	55
4.2.1	System model . . . . .	56
4.2.2	Placement mechanism . . . . .	57

4.2.3	Experiment, metrics, baseline . . . . .	57
4.2.4	Results . . . . .	58
4.3	Summary . . . . .	60
<b>5</b>	<b>Decentralized Estimation of Geographic Video Views</b>	<b>63</b>
5.1	Motivation and general system architecture . . . . .	65
5.1.1	System model . . . . .	65
5.1.2	Background: Self-organizing overlays . . . . .	67
5.1.3	A decentralized placement protocol . . . . .	68
5.2	Mignon: Fast Sum Estimation . . . . .	71
5.2.1	Trapezoidal rule. . . . .	72
5.2.2	Gregory-Newton Interpolation. . . . .	73
5.2.3	Polynomial Least Squares Fit. . . . .	73
5.3	Mignon Evaluation . . . . .	74
5.3.1	Accuracy Comparison . . . . .	76
5.3.2	Mignon Sensitivity Analysis . . . . .	78
5.4	Discussion . . . . .	86
5.4.1	Influence of Sample & Collide . . . . .	87
5.4.2	Convergence speed . . . . .	87
5.5	Summary . . . . .	88
<b>III</b>	<b>Conclusion</b>	<b>89</b>
<b>6</b>	<b>General Conclusion</b>	<b>91</b>
	<b>Bibliography</b>	<b>97</b>



# List of Figures

3.1	Popularity map of the most viewed video of our data set <i>Justin Bieber - Baby ft. Ludacris</i> , as provided by Youtube. . . . .	35
3.2	Frequency distribution of tags . . . . .	39
3.3	Distribution of the number of tags per video . . . . .	39
3.4	Median number of views vs number of embedded tags . . . . .	40
3.5	Cumulative distributions for video's views by number of tags . . . . .	40
3.6	Video's entropy vs. its number of views . . . . .	42
3.7	Video's entropy vs. its number of embedded tags . . . . .	42
3.8	CDF of videos and tags vs entropy . . . . .	44
3.9	Heatmap of the mean views for every occurrences of a given tag, versus the mean entropy of every occurrences of that tag. Mean showed as a dashed line. . . . .	44
3.10	Videos associated with the tag 'bollywood' tend to be viewed mainly in India, USA and UK. . . . .	45
3.11	Videos associated with the tag 'favela' are mostly viewed in Brazil . . . . .	45
3.12	truc . . . . .	46
3.13	Tag entropy versus video entropy . . . . .	46
4.1	CDF of prediction accuracy tag-based placement vs baseline . . . . .	52
4.2	Prediction accuracy vs video views . . . . .	53
4.3	Prediction accuracy vs video entropy . . . . .	54
4.4	A tag-based placement strategy consistently improves the system's global hit rate by about 6%, independently of the number of copies per video. . . . .	59
4.5	Hits and misses for the top 6 countries for various $R$ . . . . .	59
4.6	Performance comparison between tag-based approach and the baseline . . . . .	60
5.1	Using tags to predict users' affinity with a new video . . . . .	66
5.2	Placing new videos based on aggregated affinity . . . . .	66

5.3	A self-organizing overlay . . . . .	67
5.4	Overlay Architecture . . . . .	68
5.5	Peer-to-peer neighborhood optimization . . . . .	68
5.6	Interest curve for MovieLens(a) and YouTube(b) datasets. Black vertical lines represent KNN and RPS samples. . . . .	75
5.7	Evaluation of the error for both datasets MovieLens and YouTube . . .	76
5.8	Evaluation of the standard deviation for both datasets MovieLens and YouTube . . . . .	77
5.9	Fast Decentralized Area Estimation using the trapezoid rule in the MovieLens dataset(a) and YouTube dataset(b). . . . .	78
5.10	Influence for the Trapezoid Rule of varying the size of the KNN(a) and RPS(b) for the MovieLens dataset. . . . .	79
5.11	Influence for the Trapezoid Rule of varying the size of the KNN(a) and RPS(b) for the YouTube dataset . . . . .	80
5.12	Fast Decentralized Area Estimation using Polynomial Least Squares regression in the MovieLens dataset(a) and YouTube dataset(b). . . . .	81
5.13	Influence for the Polynomial Least Squares regression of varying the size of the KNN(a) and RPS(b) for the MovieLens dataset. . . . .	82
5.14	Influence for the Polynomial Least Squares regression of varying the size of the KNN(a) and RPS(b) for the YouTube dataset. . . . .	83
5.15	Fast Decentralized Area Estimation using Gregory-Newton interpolation in the MovieLens dataset(a) and YouTube dataset(b). . . . .	83
5.16	Influence for the Gregory-Newton interpolation of varying the size of the KNN(a) and RPS(b) for the MovieLens dataset. . . . .	84
5.17	Influence for the Gregory-Newton interpolation of varying the size of the KNN(a) and RPS(b) for the YouTube dataset. . . . .	85
5.18	Details of the Gregory-Newton interpolation with different RPS sizes in the MovieLens dataset. . . . .	86

# List of Tables

3.1	Popularity vector of the map of Fig. 3.1 (excerpt) . . . . .	35
3.2	The 10 most frequent tags . . . . .	38
3.3	The 10 most viewed tags (worldwide) . . . . .	38
3.4	The most viewed tags for various countries . . . . .	43
3.5	Top 3 Videos (views) containing <i>pop</i> . . . . .	43
3.6	The 5 tags with the most (above) resp. least (below) entropy (for #oc- currences > 100) . . . . .	44
3.7	Top 5 countries (views) for <i>bollywood</i> . . . . .	45
3.8	5 top countries (views) for <i>favela</i> . . . . .	45
3.9	Top 5 countries (by views) for <i>pop</i> . . . . .	46
4.1	YouTube traffic share according to Alexa . . . . .	51
5.1	Mean error percentage for various size-estimation errors, for Gregory- Newton interpolation(a) and Trapezoidal rule(b). . . . .	87



# **Part I**

## **Context**





# Introduction

## Contents

1.1	Motivation . . . . .	4
1.1.1	From assessment to research . . . . .	4
1.1.2	The rise of User-Generated Content . . . . .	7
1.2	Problem statement . . . . .	9
1.2.1	Tags as pertinent metadata . . . . .	11
1.2.2	Tags as reliable indicator for predicting consumption patterns	12
1.2.3	Decentralized predictive placement system . . . . .	13
1.3	Contributions . . . . .	14
1.3.1	Metadata analysis of a UGC service . . . . .	15
1.3.2	Predictive power of tags . . . . .	15
1.3.3	Distributed prediction mechanism . . . . .	15
1.4	List of Publications . . . . .	16
1.5	Organization of the document . . . . .	17

This thesis gathers and presents the most important part of the research we carried out over the last few years on decentralised storage systems. In this thesis, we argue that user-generated meta-data in fully distributed user-generated-content systems constitute a reliable source to determine where uploaded content will be consumed in the future. This makes it possible to predictively place this content close to future consumers, thereby maximizing the suitability of proximity caches.

## 1.1 Motivation

### 1.1.1 From assessment to research

A human society is defined by the sum of the interactions between the human beings who compose it, and thus plays a key role in the evolution of the human species. The Internet has fundamentally changed how humans communicate, and these changes are frequently described on the same level as the apparition of language, writing, or printing [52]. Like these media, the Internet has modified the range of communication possibilities previously offered to societies, and has defined an age where information starts growing and spreading with greater momentum. This medium where everyone can contribute represents today an essential pillar of the information age. This is why every interrogation on the Internet is considered central.

The emergence of this powerful tool has been noticed by political leaders who see the Internet as a possible threat to traditional institutions and to sovereignty itself [51]. For decades, governments and agencies of various countries have tried to legislate on Internet, adapting it to their own political patterns of governance. As an example, a list of the various French law projects and proposals can be found on the website of the Assemblée Nationale.<sup>1</sup>

The way Internet was designed, without any geographically delimited borders, flattening to the same importance the ability of each one to express itself and to be heard after decades of dominant position of the traditional medias and governments, has been considered as a possible threat to their abilities to properly govern a people. As for example, François Fillon, former French Prime Minister and before Minister of Posts and Telecommunications, declared in 1996 about the Internet :

---

<sup>1</sup><http://www2.assemblee-nationale.fr/>

It's not about reconsidering the freedom of use of these new networks, but we should not let alone develop itself a digital "far west" where our laws will not be applied. <sup>2</sup>

Notably, he tried via a legal amendment to the Law of Regulation of Telecommunications to make the Internet Service Providers penally responsible of the content of the services that haven't been approved prior publishing by the French Superior Council of Audiovisual <sup>3</sup>. This meant that anyone wanting to publish content via a website or any other medium, has to have this content approved by a state sponsored but independent public authority whose members are appointed by the government, following an undisclosed list of acceptance rules. Which in terms of nation-wide censorship is quite something. All these various attempts to control and censor are very common these years, and the public opinion is no more surprised when a country disconnect itself from the Internet to avoid its citizen to communicate on the country's events like it happened in Egypt in January 2011 [19] or in Syria in 2012 [55], or when a government agency, such as the GCHQ, by decades-long practices of mass surveillance impinge on Article 8 of European Convention of Human Rights and goes well beyond its legal framework [63].

There is another rising concern about Internet actors, and this one is directly linked to our online privacy. It concerns the control of the data consumed and created by each Internet user. Back in 2006, Microsoft was the only tech-related company to presides in the top ten ranking of the most market-valued companies in the world. As of today, the third quarter of 2016, the top five spots are taken by Apple, Alphabet via Google, Microsoft, Amazon and Facebook, all fives being tech companies, with four out of five handling tremendous amount of user personal data, and two of them having their entire business model relying on data. Google revenues solely accountable to ads, growth from 45billions USD in 2014 to 52billions in 2015<sup>4</sup>.

Data has now become highly valuable. One of the reasons is that as both software and hardware infrastructures where becoming more and more based on commodity products, the value of the data has grown incredibly, in correspon-

---

<sup>2</sup><http://discours.vie-publique.fr/notices/963223000.html>

<sup>3</sup><http://www.conseil-constitutionnel.fr/conseil-constitutionnel/francais/les-decisions/acces-par-date/decisions-depuis-1959/1996/96-378-dc/decision-n-96-378-dc-du-23-juillet-1996.10818.html>

<sup>4</sup><https://www.sec.gov/Archives/edgar/data/1288776/000165204416000012/goog10-k2015.htm>

dence with an age where we have never generated so much data, as demonstrated by Hilbert et al. [37]. Large and relevant datasets are now extremely valuable, and this value resides in the various possibility they offers such as providing content recommendation, price optimization and accurate ad targeting, or providing actionable insights and predicting models. They can improve categorization or semantic analysis, and as seen recently help to create and train better artificial intelligences.

However, these large and relevant datasets are hard to build, contributing to their value, and any big tech company who wants to build one have to find a way to attract millions of users in order to collect their informations and interactions. That is what, among others, Alphabet and Facebook have achieved. They strategically attracts customers through free services for day to day usage, and collects each action, each metadata on each content. They build tremendously large databases and analyse or monetize them, and the fact that private companies collect huge amount of personal data in order to store, analyse and trade it raises important questions. Data has become a resource, on a similar level than raw material, and the management of this resource from the collection to the exploitation has now become primordial, with some private actors having an heavier role than states, as reminded by Stephane Grumbach <sup>5</sup>.

This assessment pushes towards the necessity to develop alternative models, to propose solutions than can cope with the actual flow of generated and consumed data, and in the same time prevent any form of control over the traffic or the data that could come from any actor. One of the possible ways to achieve that is to propose alternative models to the centralisation of data and services around these large privately-held companies.

In an effort to guarantee freedom of circulation for the data and systems being resilient to censorship and infrastructure control, previous studies have been focused on developing distributed and efficient alternatives solutions to the centralized ones.

My PhD thesis takes place in this dynamic setting.

---

<sup>5</sup>[http://www.lemonde.fr/idees/article/2013/01/07/les-donnees-puissance-du-futur\\_1813693\\_3232.html](http://www.lemonde.fr/idees/article/2013/01/07/les-donnees-puissance-du-futur_1813693_3232.html)

### 1.1.2 The rise of User-Generated Content

When we mention big players of the actual Internet, it's impossible to not associate them with video content. Whether it be Alphabet via its subsidiary YouTube, Netflix, Twitch or Amazon Video, these services are handling a tremendous amount of downstream internet traffic as pointed by the various Sandvine reports for Europe, Asia and Americas [9], through millions of hours of content delivered daily [11, 1, 10]. Among them, Amazon Video and Netflix are services streaming to their users the content of their choice, this content having being self produced by these services or contracted from a media company. On the other half, TwitchTV and YouTube manage a content which is produced in majority by the users of these services. These type of services are called User-Generated Content services and are among the most popular online platforms<sup>6</sup>. The original role of a User-Generated Content service is to collect content under any form such as chats, podcasts, digital images, audio files, video files and other media that was created by any of its user, and make them available to every user of the platform. These services could be collaborative, where users interacts between each others in order to create, manage and improve a specific content. The free online collaborative encyclopaedia Wikipedia is the flagship of this type of services. The most represented type of service is the non-collaborative one, in which a given content has one single creator, responsible of its upload on the platform and of its management. These platforms can charge users to access the various contents produced by other users, like Meetic, or be free of charge like Flickr. Among these categories, the most notorious User-Generated Content services are the ones dealing with video content such as Twitch or YouTube, with the first reporting 9.7 millions of active users every day watching an average of 106 minutes of videos daily [2], and the second reporting hundred of millions of hours of videos accessed by users every day [1]. These User-Generated Content (UGC) services have grown extremely fast over the last few years, both in terms of involved users and volume [8, 1].

The resulting traffic now accounts for a substantial proportion of the world's network usage with YouTube accounting up to 21.16% and 24.64% of the aggregated traffic respectively in Europe and Asia, and up to 34.70% for Netflix in North America [9]. Storing, processing, and delivering this amount of data poses a constant engineering challenge to both UGC service providers and ISPs. One

---

<sup>6</sup><http://www.alexa.com/topsites/global>

of the main difficulties is the sheer number of submissions these systems must process and their staggering growth rates, YouTube for example reported in 2014 receiving 100 hours of video every minute as opposed to 300 hours every minute in June 2015 [1], most of which need to be served to niche audiences, in limited geographic areas [18, 40, 61].

The recent expansion of Netflix on the European market brings also his share of traffic in the infrastructures of the continent. Its presence being noted by its entrance in the European market in January 2012. Although available at the beginning in only few countries, Netherlands and Nordics as well as the British Isles where it accounted for over 20% of peak traffic on several networks, it was still absent of the top ten applications producing the most aggregated traffic at the end of 2012 [7], but in only one year, it growth to reach the fifth place at the end of 2013, representing suddenly 3.33% of the aggregated European access on fixed access lines. This rapid growth is noticeable when we take in count that it took Netflix four years to achieve that level of share in the United States [8].

In order to support this growth, current services typically exploit private datacenters owned by large companies such as Alphabet, Sony and Amazon. These datacenters are further augmented with Content Distribution Networks (CDNs) (such as Akamai or Cloudfront) and caching servers positioned at points-of-presence (PoP) within the infrastructure of Internet Service Providers (ISPs) [34].

This approach tends to favour big players, and to concentrate the industry in the hands of a few powerful actors. For several years now, both academia and practitioners have therefore sought to explore alternative designs to implement social online services in general, and UGC video services in particular. One strategy espouses a fully decentralized organization [13, 16, 40, 45, 53], in which each individual user (through her computer or set-top box) provides storage capacity and computing resources to the overall system, eschewing the need for any centralized element, either hosted in the cloud or on replicated servers.

Many individual services have in particular been proposed to move this vision forward including storage [45, 54, 58], indexing [22], queries [14, 41], recommendation [15, 16, 17], caching [31], and streaming [20, 32]. Because these services are highly distributed, they cannot rely on any central coordination component to organize themselves, for instance using a top-down or fixed hierarchical organization. Instead they typically exploit decentralized adaptation strategies to shape their structure and behaviour according to the demands placed on them,

and the resources at their disposal. As such they are a very good example of modern self-organizing distributed systems.

To ensure their scalability, most of these services primarily rely on *limited interactions* (e.g. with a small set of neighbouring nodes) and *local information* (e.g. users profiles, bandwidth, latency, tags). The use of local information is one of the key reasons why these services scale, as they do not need to gather and aggregate data distributed over a large number of nodes to progress, a costly process in fully decentralized systems. Enforcing a strong focus on locality, however, constrains the range of decisions that can be taken by individual nodes, and their ability to adapt to more complex phenomena occurring at a global scale. In an attempt to address this limitation, my researches have been focused on the particular problem of *global predictions* in large-scale decentralized systems, with an application to the placement of videos in a decentralized UGC service.

In this thesis, we argue that in order to propose a neutral, efficient and decentralized alternative to big players UGC services, one important point is to be able to anticipate where and in which proportions of an uploaded content will be consumed in the future and then place it accordingly close to its future consumers, in order to push the traffic to the edges of the decentralized UGC systems. More precisely we consider the problem of a newly uploaded video that must be stored and replicated within a peer-to-peer system in the countries where it is more likely to be viewed. One way to achieve that is the one highlighted by my works, by non-intrusively leverage the metadata defined by the uploading users, in order to build a decentralized predictive content placement protocol, for further integration in decentralized UGC services.

## 1.2 Problem statement

The trade-off between a decentralized UGC service focusing on *local information* and the ability for each user to have a *global knowledge* of the activity of the other nodes often results in an increasing amount of traffic generated *inside* the network. As stated before in Section 1.1, in UGC services, an important share of the various contents tend to be consumed by restricted audiences in limited geographic areas. Considering this observation, it seems important to maximize the traffic in these delimited geographic zones, in order to avoid obstructing the core of the network. This not-so-new concept aiming to draw the consumer closer



to the producer without intervention of a central node or having to follow third-party determined path, has been found recently at the heart of efforts such as *Edge computing*, or *Fog computing* in our particular case.

In this context, recent trends towards full scale data analysis showed very promising leads and results when consumption habits are analysed to predict the future needs, or when past schemes analysis can reveal patterns announcing upcoming events. These concepts have been profusely used and developed in the establishment of accurate recommendation engines [56, 47, 44, 28], among other examples. The most common practices consist in collecting data related to the consumption of consumers, and link for each type of content the concerned audience and their interests towards them. Thus, a newly available product can, with its own characteristics, be proposed predictively to the consumers that might be interested in it. In our case, this solution embodies three different problems.

- First, identify what are the markers used in UGC services by the producers to categorize a content that are linked to the consumption habits of the consumers. We want to analyse and see if among these markers if one can be related to a geographical pattern of consumption of the content it is linked to.
- Second, if this marker is reliable enough to be related to the consumption of the content that it marks, we need to go further and check if we can rely on this same marker to predict where a given content will be consumed during her available time, right at the moment when the content is produced and the markers have just been set. In other words, we need to verify that we can quantify the future interests of a delimited geographic area towards a given content freshly produced, just by looking at this particular marker and the ability to build an efficient content placement mechanism based on the observation of the marker.
- Third, we need to allow any node to calculate the aggregation of the interests of any user comprised in any delimited geographic area on a time and efficient way, without the use of any central element or super-peer node, in order to implement this placement mechanism in a fully decentralized way.

### 1.2.1 Tags as pertinent metadata

Along the years, many markers have been used in various UGC services, in order to help to categorize and to evaluate content. This metadata can be attached to the content to which it relates by three manners: *(i)* by being set automatically by the service itself, *(ii)* by the consumers or *(iii)* by the producer. When being set by the service itself *(i)*, it can be for organisational purpose such as automated categorization or to indicates the appreciation of the users such as the number of times a content has been accessed. Automated categorization rely more and more on an automatic analysis of the uploaded content to determine the most appropriate markers to attach via graphical analysis or textual parsing for examples. As for the rest, it can be an auto incremented counter of visit on a web page or the number of views gathered by a video. When the metadata is set by the consumers themselves *(ii)*, we find again the two categories, where consumers can add textual information about a content, generally in form of tags like in MovieLens<sup>7</sup> where this system is used to add an external textual information to help describing and relate any content. Or it can also be of the second form, especially when users give an appreciation notation to a content, in order to share their appreciation of it. This second form is very common and is used to gather the note given by any willing consumer for any video in Netflix<sup>8</sup>, in order to present to any consumer of a content the aggregation of notes given by others, or on Reddit<sup>9</sup> by a system of upvote/downvote to promote a content to the front page, or on contrary to drive a disliked content to oblivion. The third way to attach metadata to content is through the producer *(iii)*. When one upload content in UGC services, especially video ones, it usually comes with a title and a short textual description aimed to orientate the other consumers and provide keyword for search engines, it can also be a suggestion of category or a set of tags related to the content.

This last way to generate metadata attached to a content is the more interesting for us, as it is the least intrusive one. Actually, the public display of this metadata is the choice of the producer and only of the producer, we do not need to know what the uploaded media is made of, or our service to run intrusive analysis, the metadata as been willingly attached by the producer in full knowledge. More precisely, we look closer to the tags that are set by the producer, as this free format of description is designed to embody a concise and precise description of

---

<sup>7</sup>[www.movielens.org](http://www.movielens.org)

<sup>8</sup>[www.netflix.com](http://www.netflix.com)

<sup>9</sup>[www.reddit.com](http://www.reddit.com)

the content, and is a widely adopted self-description mean. Our first problem will then be to determine if these freely applied markers attached by the producer of a content can be related to geographic consumption pattern of the content they are attached to.

### **1.2.2 Tags as reliable indicator for predicting consumption patterns**

Once we identified an adequate marker that reflects the geographic consumption of the content it is attached to, we need to verify that it can fulfil our hopes. We want that marker to embody the future geographic consumption pattern of the content it is attached to. We need to verify whether or not, video content popularity in our case is reflected by a video's views and their repartition among the various geographic areas can be inferred from its tags. Once this potential will be verified, we will analyse the efficiency of a placement mechanism in UGC systems that will rely on the previous observations. This mechanism should leverage the knowledge on content consumption obtained from the marker observation to maximise the closeness between a content placement and the users that will consume it. Among the various obstacles to that open possibility, three are the most challenging ones. The fact that any tag can be attached to any video, regardless of the nature of the content is a challenge that could not be addressed directly. The study of the prediction potential of any tag has to be a study of consumption. A grammatical study or a lexical analysis of the tags seems too immutable for us and does not take in account the actual trends in the use of the various possible tags. An efficient metric based on the previous use of each tag must be found in order to discover if tags can help predict the geographic consumption pattern of their attached content.

Then in a second time, the quantification of the efficiency of this prediction must be also suitably measured. This step represent the final piece of answering the question about the promises of relying on tags for our researches. If we can find a way to measure the predictive potential of tags on which we can rely on, we can then prove that we can safely build reliable systems on this ground. Discovering this efficient metric and measuring the accuracy of the prediction is thus important. Actually, any placement system, proactive or not, comes with an immediate cost higher than keeping the content on the storage space of the producer, thus, quantifying the efficiency of the prediction directly impact the

efficiency of a predictive placement system and thus counter balance its own cost.

And then, in a final step, we will have to propose a predictive placement system. After having proved or not the predictive potential of tags, it will then be time to introduce a placement system using our previous observations as leverage. We will have to propose and evaluate a system that can deduce from the sole observation of the tags attached to any produced content, where in the world, and in which proportions this precise content will be consumed. This proposition will have to meet actual architectures, in order to propose a credible mechanism that can be integrated to current UGC services, and not make assumptions on hypothetical and unachievable situations. Further more this system should be thrifty and not demand enormous level of knowledge to be able to reach a high accuracy in its predictions.

### 1.2.3 Decentralized predictive placement system

One final key difficulty when applying predictive techniques to decentralized systems is the need to decentralize the prediction itself. When the two firsts problems will be tackled, we will have a functioning concept. We will have a predictive content placement mechanism, able to accurately predict where and in which proportions a newly uploaded content will be consumed in the future, only by looking at its tags attached to it by the producer, and so be able to place the replicas close to its future consumers, in order to minimise congestion. However, having to query a centralized actor for each content upload will be self-defeating, counter-productive, and besides neglecting privacy issues, will be ignoring the full potential of such content placement system. Now, decentralizing such system is equivalent to decentralize the knowledge that we can learn from such a tag study explained previously. This problem comes in two different parts.

First, we have to explore a way for each node to be able to ask any node about its interest towards a given set of tags. Because each node cannot keep a record of the consumption habits of every other user, we have to be able to realize our tag study in real-time. It means that every user when wanting to upload a produced content, will have to perform the tag analysis mentioned before on the tags she want to attach to her content. We want of course to avoid oblivious solution like flooding the network of queries, using a token-ring approach or any solution that is not fully scalable and cheap. Again, we want a full decentralized system, implying in our case that each node is running the exact copy of the system than

any other node, thus also eliminating the solutions based on proxying the interests requests or the ones based on super-peers. In addition to impacting as little as possible the load of traffic in the network, the query operation has to be as fast as possible and be able to target a precise geographic region or else the responding nodes should incorporate a way to geographically taint their answers.

Second, the manner how the nodes communicate their interests towards a given set of tags to the producer has to be extremely concise. Our ideal decentralized system has a granularity of end-user machines or set-top boxes, implying various hardware characteristics and bandwidth capacities. We cannot afford for each node uploading a content to receive the entire previous consumption profile of each other node in the network. The associated costs in term of overall traffic, free memory space and computing time for the producing node would be unacceptable. As for the number of queries needed to interrogate each node in each geographic region, the volume of the answers received by the producer should remain extremely low. We need then to develop a solution allowing each answer to be as lightweight as possible and in the same time embodying enough informations for the producer to determine the geographic consumption patterns. This innovative solution should not focus on condensing a bulk load of data by any compressing way, but on the contrary try to be as small and efficient right at the source, leading the way to future developments and upgrades.

This tripartite problem reminds us of the three problems presented in the following thesis. Each of these three parts will be tackled by a creative solution and a rigorous analysis in the rest of my thesis, introducing my work along these three years. My contributions are presented in the following section, making match face to face each solution and the related publication.

### **1.3 Contributions**

In this thesis, we introduce a predictive content-placement mechanism designed for User-Generated Content services in a distributed environment. To define such a mechanism, we started by identifying three problems. The first consists in finding, among the various markers available in the metadata attached to content in UGC services, one that can be related to the geographical consumption pattern of the content it is attached to. The second one is complementary to the first, and aims at verifying that once this marker has been found, we can

use it to predict the future consumption pattern of the attached content. The third one consists in decentralizing this prediction mechanism so it can be incorporated into a distributed storage system. This thesis therefore proposes three contributions, each representing a possible solution to one of these problems.

### 1.3.1 Metadata analysis of a UGC service

Our first contribution [24, 26] analyses the tags attached to the various videos that are part of a large YouTube dataset. We focus on the existing relationship between the geographic distribution of views of a given video and the geographic distribution of views of the tags associated with it. We use Shannon entropy as a metric for the geographic spread of the views associated with a video or a tag. In this first part, we highlight that a large proportion of tags and videos have a view distributions concentrated in very limited geographic areas. We also reveal that low-entropy videos are linked to low-entropy tags and vice versa. These two observations suggest that tags can be used to estimate the geographical areas and the countries associated with the views of an uploaded video.

### 1.3.2 Predictive power of tags

The second contribution [24, 27] goes one step further and tackles the question of the predictive power of tags. While our first contribution highlights a link between tags and the geographic distribution of views for a given video, this second part analyses the possibility to rely on tags to predict the consumption pattern of a video. This prediction must take place right at the upload when the producer has decided which tags to attach, before the content starts being consumed. Through this contribution we demonstrate that tags can predict the distribution of views with reasonable accuracy. This makes it possible to integrate this prediction model into a proactive placement system, that automatically stores newly introduced content close to its future viewers.

### 1.3.3 Distributed prediction mechanism

Our last contribution [23, 25] introduces Mignon, a novel protocol that leverages the two previous contributions in a decentralized context. With our first two contributions, we are able to predict accurately where a video will be consumed in the future. However, each uploading node needs to compute these pre-

dictions independently either by relying on a central database that keeps track of the global consumption, or by maintaining complete knowledge of the view distribution of each video and each tag. These two cases do not match with a distributed, efficient, and scalable environment. One of the possible answers to this problem resides in Mignon. Mignon tackles the problem of estimating the viewing potential of a video for each country in the world, in a fast and efficient way with very little overhead in terms network traffic.

## 1.4 List of Publications

- Stéphane Delbruel and François Taïani. From views to tags distribution in youtube. In Romain Rouvoy, editor, *Proceedings of the Middleware '14 Posters & Demos Session, Bordeaux, France, December 8-12, 2014*, pages 21–22. ACM, 2014.
- Stéphane Delbruel and François Taïani. Géodistribution des tags et des vues dans Youtube In *Conférence d'informatique en Parallélisme, Architecture et Système. Compas'2015.*, Lille, France, June 2015.
- Stéphane Delbruel, Davide Frey, and François Taïani. Decentralized view prediction for global content placement. In *Proceedings of the 14th International Workshop on Adaptive and Reflective Middleware, ARM@Middleware 2015, Vancouver, BC, Canada, December 7-11, 2015*, pages 10:1–10:3. ACM, 2015.
- Stéphane Delbruel, Davide Frey, and François Taïani. Exploring The Use of Tags for Georeplicated Content Placement. In *IEEE International Conference on Cloud Engineering IC2E'16*, Berlin, Germany, April 2016. **Best Paper Award**
- Stéphane Delbruel, Davide Frey, and François Taïani. Mignon: a Fast Decentralized Content Consumption Estimation in Large-Scale Distributed Systems. In *16th IFIP International Conference on Distributed Applications and Interoperable Systems DAIS'16*, Heraklion, Greece, June 2016.

## 1.5 Organization of the document

This manuscript comprises 6 chapters, organized in three main parts (*Context*, *Thesis Contributions*, and *Conclusion*).

- The next chapter, Chapter 2, presents the related work in the field of geodistributed content delivery, and decentralized storage systems, highlighting the challenges required to realized fully decentralized UGC services.
- In Chapter 3, we analyze a rich Youtube datasets containing information about the geographic distribution of video views, and investigate the relationship between the tags attached to videos, and the view distribution of videos.
- Chapter 4 builds on the results of Chapter 3, and explores how tags might be used to predict the view distribution of a new video, and whether this ability might be useful in a largely decentralized UGC service.
- Chapter 5 extends the approach proposed in Chapter 4 and presents an architecture and a novel mechanism to realize a proactive placement prediction in a fully decentralized setting.
- Finally, Chapter 6 concludes and discusses future work.





## Related Work

### Contents

2.1	Distributed storage systems . . . . .	20
2.1.1	Freenet . . . . .	20
2.1.2	PAST . . . . .	22
2.1.3	Storj . . . . .	23
2.2	Content distribution . . . . .	24
2.2.1	Hybrid CDNs . . . . .	24
2.2.2	Content Delivery for alternate systems . . . . .	25
2.3	Metadata in UGC services . . . . .	25
2.3.1	Consumption patterns . . . . .	26
2.3.2	Geographic properties . . . . .	28
2.4	Summary . . . . .	29

The theme of peer-to-peer storage systems emerged way before the appearance of the big data era we are living today. Data was not as big and valuable as today, so the immediate goal was to explore alternatives to the different centralized systems and at the same time continue to promote the research towards a decentralized Internet, in the right path of its original spirit. Although some initial peer-to-peer systems adopted a semi-centralized architecture with a central server proxying requests [60], the interest of peer-to-peer lies in the ability of user machines to act both as clients and as servers [57]. For the rest of this thesis, we will thus exclusively refer to the fully decentralized ones every time we mention peer-to-peer systems.

## 2.1 Distributed storage systems

The first generation of peer-to-peer applications, like the Gnutella network [57] has been extensively studied and commented over the years, and particularly criticized for its various flaws, Gnutella being too greedy on the bandwidth thus limiting its scalability and Napster relying partially on central elements. We would rather start analysing from the second generation of decentralized architectures emerging at the beginning of the years 2000s, such as Freenet [21] and PAST/Pastry [59], and introducing clever features that new generation of decentralized systems still rely on as of today. Chronologically, the first one to appear was Freenet at the beginning of the year 2000.

### 2.1.1 Freenet

Freenet [21] is an application based on an adaptive distributed network of nodes, enabling the publication, replication and retrieval of data, with a strong focus on the anonymity of both readers and authors. Each node is responsible for maintaining two separate elements: first, a local datastore which it makes available both for reading and writing to the network, and second, a routing table containing addresses of other nodes and the keys they are thought to hold. Keys are used to uniquely identify a file. Nodes are organized upstream and downstream as a proxy chain. When a user wants to retrieve a file, she first looks it up on her local datastore, and if not present, forwards the request to the host found in her local routing table having the closest key to the one requested. The request is then oriented hop by hop to nodes having a closer key in their database, until

reaching the node who actually store the targeted content. Loops and dead ends in the routing process are handled with backtracking but nonetheless a request can still fail to find the proper target when exceeding her hops-to-live limit. The failure is then reported to the requesting node. When the targeted node is found, the data is returned to be cached in the requesting node, taking for that the shortest path among the routes used in the requesting sequence, and caching replicas along the way. This mechanism helps to improve the quality of routing over time by specializing nodes in locating sets of keys similar to the one attributed to them in the table of their neighbours, and also tend to become similarly specialized in storing clusters of files having similar keys.

Adding new content to the system follows a similar strategy as the one used for routing content. The user first computes a file key for the content she wants to insert. The key will be propagated as if it was a request along the mechanism explained before, being directed towards the node having more and more similar keys. When the key collides with one already present key in the local datastore of a node, the key is returned back to the uploader node in order to let her try again with using a different key. When the insert request reach the hops-to-live limit without a key collision being detected, an "all clear" result is returned to the uploader node, being this time a successful result. The uploader then send the data to insert, which will be propagated from node to node following the path established by the insert query, and the data will be stored in the local datastore along with the key in the routing table, of each node along the way.

The main drawback of Freenet derives from the use of keys based on hashes of the descriptive strings attached to a file. This yields to cluster lexicographically close hashes rather than subjects. As the bright part of this method is by disseminating around the network similarly described content will ensure that the failure of a part of the network will have lessened chances of rendering unavailable all the contents associated to given subject, it does not allow the overall routing scheme to be as nimble as we can get when the consumption traffic is pushed to the edge of the network like in a similarity-driven overlay. For the sake of clarity, we do not detail how FreeNet handles adding a new node in the network or how it handles the problem of finite storage capacity.

### 2.1.2 PAST

PAST [59] is a large-scale decentralized storage utility, and is very interesting due to its strong focus on storage management and caching mechanisms. It is built upon Pastry [58], a well-documented scalable distributed object location and routing overlay, designed to operate in a peer-to-peer network. Pastry has a slightly different routing mechanism than Freenet detailed before, and runs by routing an associated message towards the node whose nodeID is numerically closest to the 128 most significant bits of the fileID, which one is obtained by computing the SHA-1 cryptographic hash function of the name of the file, the public key of the owner and a randomly chosen salt. A PAST node can interact upon three different ways. When a user, by means of a node, sends an insert request, the fileID is computed and the client's storage quota is amputated of the file size multiplied by the defined  $k$  number of replicas. The fileID is used as a destination to route the file via Pastry, and when the first node among the  $k$  required is reached, it checks the integrity of the message, accepts to store one replica and forward the request to the others  $k-1$  nodes. The second type of interaction is for an user to retrieve a given content. The request message is routed with the fileID as destination. As soon as the message reaches one of the  $k$  nodes storing the file, the requested content is sent back to the requester. A third and last interaction is the reclaim one, related to how via a reclaim certificate the owner of a file can reclaim its associated storage space, nevertheless it does not guarantee that the file is no longer available.

One interesting point in PAST is how the system handles the various problems linked to content distribution. PAST through its cache management aims to minimize client access latency, to maximise the query throughput and to balance the query load in the system. The  $k$  replicas of a file are handled by PAST mainly for reasons of availability, however this number of replicas has to increase for a highly popular file, in order to sustain its lookup load and minimize client latency. Further more, it's important to store a copy near each cluster of interested users. In order to create and maintain such additional copies, cache management in PAST nodes use the "unused" portion of their advertised disk space. Cached copies can then be evicted and discarded at any time, it's typically happening when a node stores a file or one of its replicas. In order to make room for the new storage request, the concerned node will enforce the GreedyDual-Size cache policy. Cache insertion on the contrary happens during a lookup or insert phase, and

concerns the nodes from whom the request with the file is routed, as long as their free portion of their own local storage qualify for the different requirements. The main drawback of this scheme is that the caching performance decreases when the storage utilisation of the system increases. Nonetheless, the brute and blind caching policy of PAST brings the advantages of strong persistence and reliability.

### 2.1.3 Storj

One of the latest and most interesting decentralized storage network is Storj [64]. Storj's goal is to be an open-source decentralized cloud storage network allowing each user to transfer and share data without reliance on a third party data provider. It shares with the two systems presented previously, and especially Freenet, the goal of getting rid of data providers serving as trusted third parties, on whom cloud storage on the Internet has come to rely almost exclusively. However it differs in many points, each one being the direct consequences of years of research and improvement between the first propositions as the ones explained earlier, and the weaknesses of a trust-based model as proposed mostly by cloud storage companies. The firsts points are related to trust and privacy between a user and the overall system. Storj handles files by their content via a hash, and enforces end-to-end encryption in order to protect the data in transit and on the devices not controlled by the user. When uploaded, a file is split into shards – an encrypted portion of a file with a constant size – and these shards are disseminated along the network, ensuring no node will have a complete copy of the file, as long as its size is greater than the standardized size of a shard. The metadata of each file is stored in a Satoshi-style blockchain, that is to say the location of each shard, the file hash used for indexing and her Merkle root. To ensure the integrity of each shard stored on each storage space of each user, audits are done using Merkle trees [49]. As a blockchain is a public ledger, it's a very good tool to achieve a distributed consensus on shards location and dissuades tampering attack on files. The last main point in which Storj differs from the earlier distributed storage systems, is its embedded mechanisms designed to cope with redundancy and so availability. Instead of blindly distributing a fixed number of replicas across the network, the shards are stored using a K-of-N erasure coding scheme with multiple nodes. The client may choose K and N to achieve a balance of robustness and costs. In case of a number of shards from various nodes are discarded after an audit, due to fault, tampering or simple disconnection, the file

can still be reconstructed if there is at least  $K$  shards among the  $N$  produced. To restore the balance a replication process is applied whereby one of the existing copies on the network is replicated to a new node, thus the network is able to heal itself after each audit.

## 2.2 Content distribution

Besides storage, one of the main problems addressed with decentralized architectures is content distribution [13]. Delivering content to users on an efficient way has been studied for many years now. Among the main motivations, the ability to ensure content delivery, reduce the load on the content provider and the delays were at the heart of the developments of Content Distribution Networks. Today, CDNs serve a large fraction of the Internet content: text, documents, media files and live streaming media for naming a few.

### 2.2.1 Hybrid CDNs

The constant enhancing of these networks pushed to develop hybrid architectures using peers to distribute the content, and not stay fixed on a static set of caching servers allocated in various geographic locations. These hybrid CDNs are interesting for us because they give a hint on the various problems encountered when wanting to smartly deliver a content in a distributed environment.

A hint on the complexity of the current architectures has been provided by Huang et al. [39] by examining the workload of the photo serving stack of Facebook and the role of the many layers that are part of it. This architecture employs various layers, in order to provide the content as close as possible from the client. We first encounter a local cache in the browser or the mobile application of the client. This limited cache is linked to Facebook Edge, an architecture of cache servers located in various points of presence (PoPs) around the world running a FIFO cache replacement policy. If the requested content is not found in the Edge cache, the request is relayed towards the Origin Cache, a set of cache servers located in Facebook's datacenters also running a FIFO eviction policy. In the final case where the Origin Cache is not able to serve the requested content, Origin servers will fetch the content from the storage servers of Facebook, namely the Haystack.

This complex solution perfectly describes the various challenges that delivering a content worldwide close to the consumers represent. The philosophy of *enter deep into ISPs* by placing a level of cache directly into the PoPs has been well enforced by Akamai [38] and represent an effort in pushing the cached contents as close as possible to the consumers while remaining inside the perimeter of the company. Pushing the cache levels towards the machines of the consumers break the barrier of the ISP infrastructure and allows to reduce significantly the costs and the server load [38].

### 2.2.2 Content Delivery for alternate systems

The situation detailed in the previous subsection is very efficient and allows great levels of reliability, however, it lies on a wide and costly infrastructure, that not everyone can afford. One way to have an efficient delivery infrastructure without servers is to rely only on peers. One of the difficulties is then to achieve an efficient placement of content among the peers that agreed to participate, in order to maximize the proximity between the peer acting as a local cache and the consumers, thus tending towards high performance. An example of this is Flower-CDN [30] a locality-aware P2P-based content-distribution network. By relying as much as possible on peers to deliver content that would otherwise be provided only by the server, this approach comes closer to a fully decentralized system than the hybrid CDNs detailed above. In Flower-CDN, peers interested in some content cache it to later serve it other peers. They are organized in clusters via content-based overlays, and each consumer relies on a global P2P directory, based on a DHT, indexing these overlays to be redirected towards one cluster or another one function of its request. This approach allows Flower-CDN to have better lookup latency by a factor of 9 and transfer distance by a factor of 2 compared to an existing Hybrid P2P-CDN strictly based on a DHT.

This efforts indicates that when we deal with alternate, fully decentralized systems, a strong focus has to be put towards geographic locality when it is about content distribution.

## 2.3 Metadata in UGC services

The differences between the systems described above witness the evolution of decentralized storage and content-delivery systems over the year. The strong pen-



chant to secure the communication channels and be able to verify the integrity of any data without any central authority illustrates perfectly the distrust towards any decentralized system operated by a major actor, or even more the growing number of different attacks suffered by any open system as of today. But if the current systems have evolved to overcome the weaknesses of a trust-based model and reaches new highs in efficiency, there is still some domains which have been little to poorly investigated. One particularly is the strategies of placing content and its replicas right at the upload. We saw previously different approaches, with PAST placing its replicas along the routing path of the requests-replies messages, Freenet having a seemingly approach, and Storj distributing the scattering of replicas to avoid censorship or the loss of a file due to a massive disconnect from a cluster of users. However, these fixed approaches take into account only partially the evolution of viewing patterns, legacy of decades-long schemes with one producer serving multiple consumers. An important point popping from these previous reviews is that core principles of the design of a decentralized storage system are different depending how the data is produced and consumed by the various nodes.

### **2.3.1 Consumption patterns**

As the general model whereby the information produced by one or a limited set of nodes will be stored and consumed all over the network has been profusely studied of the past decades, the still recent upcoming of User-Generated Content services has brought new consumption patterns and studying them is key towards maximum efficiency. A study by Cha et al. [20] published in 2007 and relying on a publicly available dataset of YouTube gives some serious insights about the popularity life-cycle of videos and the intrinsic statistical properties of requests and their relationship with video age. They oppose the old historical standard Video-on-Demand systems where content was created and supplied by a limited number of actors, such as licensed broadcasters and production companies with marketing campaigns therefore easily controllable, to the UGC model where users are hundreds of millions of self-publishing consumers, leading to a content popularity more ephemeral and with a more unpredictable behaviour. Actually, the most representative part of UGC services as of today is based on a model where any user can at any time consume any content produced by any other user at anytime. One of the first side effects observed are content length

is shortened by two orders of magnitude and so the production time. Hence understanding the popularity characteristics is important because it can bring forward the latent demand created by bottlenecks in the system. We will detail only the main parts of the studies presented in [20] for brevity reasons, and because the legality of the uploaded content and improving centralized UGC via decentralized distribution are outside the scope of this thesis. The datasets used in their analysis comes from Daum, known as Kakao as of today, and YouTube, both UGC services. An interesting analysis is produced by observing the evolution of popularity distributions of UGCs. They first observed that the YouTube distribution seems highly skewed towards popular files, with 10% of videos presents in the dataset account for more than 80% of views, with Daum data exhibiting a very similar behaviour. With this observation in mind they delve deeper towards the observation of popularity distribution of videos for four representative categories, *Entertainment* and *Science & Technology* for YouTube, *Food* and *Travel* for Daum. All of them exhibit a power-law behaviour (a straight line in a log-log plot) across more than two orders of magnitude, but two of them have a sharp decay due to exponential cutoff. This consumption cut-off is mainly category-dependent, nonetheless most of the categories displayed a *power-law waist* with a *truncated tail* fitting best by a power-law with exponential cutoff. The truncated tail effect described here have been suspected by Gummadi et al. in their study of file popularity in P2P downloads [36] to be caused by a "fetch-at-most-once" behaviours of users. This effect could be common to UGC services due to the immutable nature of the content, having the users less likely to watch the same video multiple times. The authors then state that so far, the tail truncation for the popularity distribution of content in popular categories is affected by both the average requests by users and the number of videos in a category. Next to that study on popular content, the study shift towards the unpopular ones, constituting the so called "Long Tail" and its embedded opportunities described in Anderson's book [12]. The main goal was to identify the underlying distribution of non-popular items and which phenomena are shaping this distribution. They obtained the best fit on the distribution with Zipf with an exponential cut-off, the exponential cut-off being the most relevant to fit the truncated tail. In analysing the reasons why the distribution has this shape, few reasons have been mentioned, going from a natural shape due to the very own nature of UGC, its wide variations in quality and normally being produced for small audiences, to

other reasons like information filtering. To detail this last one, information filtering here refers to bottlenecks due to post- and pre-filtering, such as recommendation engines typically favouring a small number of popular items in the first case, and the last one being sampling bias. This study continues with the temporal evolution of the previously cited popularity distributions. As opposed to standard VoD services where content popularity fluctuation is rather predictable, UGC video popularity can be ephemeral and has a much more unpredictable behaviour. The authors observed an ephemeral popularity for young videos during a few days after the upload. While interests of users seems to be video-age insensitive on a gross scale, on a one day period, roughly 50% of the top twenty videos are recent ones, and as the time-wondow increases, the median age shifts towards older videos, confirming ephemeral popularity of young videos. When addressing the temporal evolution of the popularity, the authors found that probability of a given video to be requested decreases sharply over time. In fact, this indicates that if a video did not get enough requests during its first days, it is unlikely that they will get many requests in the future. Less than 1% of new videos make it to the top popular list, the rest having their popularity dimming over time creating a massive amount of very limited niche audiences, and their chances of becoming popular in the future being barely existing, although some few very noticeable exists due to rare and circumstantial phenomena like in this famous one <sup>1</sup>.

### 2.3.2 Geographic properties

This vast amount of niche audiences presented above could be a real challenge to distribution components of a decentralized storage systems. However, some very interesting properties of these niche have been unravelled in various works. Brodersen et al. in 2012 [18] and Huguenin in the same year [40] are two important works bringing decisive observations on which we rely. Brodersen focused his study on the relationship between popularity and locality on YouTube videos. Their investigation aimed to determine if YouTube videos exhibit geographic locality on interest, with views arising from a confined spatial area rather than from a global one. Their results depicts how despite the global nature of the Internet and with a globally available UGC service like YouTube, online video consumption is constrained by geographic locality of interest. In a first time, the

---

<sup>1</sup><https://www.youtube.com/watch?v=dQw4w9WgXcQ> – Stats

authors investigate to determine if videos of YouTube tend to experience geographic locality of interest, or rather, a uniform view popularity in the different countries. They uncover that about 40% of videos in their dataset comprised of more than 20 millions of videos have at least 80% of their views originating from a single country, this share lowering only to 70% of the views for the majority of the videos in their dataset. This strong evidence supports the claim that videos in large UGC systems tend to be viewed mostly in a limited geographic area. The temporal evolution depicts video views are more likely to have their daily view peak on their first viewing day and second day, with respectively 38% and 22% of the videos present in the dataset behaving accordingly. The daily views then decrease on a seemingly exponential form. The spatial observations related to that phenomenon highlight a spread-and-withdraw effect. When the peak of daily views is reached, these interest tends to come from a limited geographic area. Following this peak, the video start to have a much wider audience, with its views being disseminated to more and more regions. Then more the time pass, more the origins of the views tends to come from a very specific location, reaching an even greater specific locality than the one reached during the peak of focus. Huguenin [40] adds to this observations that in their YouTube dataset, less a video is popular i.e having a small number of views, and more these views will be concentrated in one or a few countries. Claiming that more a video is popular, more she tends to be equally viewed in every country. The impact of social sharing interactions on the geographic properties of a video views by Brodersen are not developed in this thesis, so does the ability of YouTube content graph to help predicts the geographic viewing pattern.

## 2.4 Summary

A large number of works have investigated content delivery architectures, from traditional Content Delivery Networks (CDNs), to P2P caching systems, through hybrid solutions combining elements of both strategies. Similarly a large number of fully decentralized storage systems have been researched over the years, demonstrating the scalability and robustness of these approaches.

Most of these decentralized storage and delivery systems however either ignore spacial locality issues, or mostly limit themselves to reactive caching strategies. This is problematic as fully decentralized approaches often cannot rely on

the amount of resources available to large centralized corporate solutions that are predominant in today's market. In order to improve the performance of alternative decentralized solutions, we therefore propose to explore how the meta-data attached to content might help produce an intelligent proactive placement of content, thus improving the overall locality and hence performance of the resulting system.

We focus in the remainder of this thesis more particularly on User Generated Content (UGC) systems delivering videos, and the tags attached to the videos to investigate this research question. We start our investigation by analyzing the correlation of tags and geographic view distribution in the next chapter, before building on the results of this analysis to propose a number of predictive placement mechanisms suited to decentralized storage and delivery systems.

**Part II**

**Thesis Contributions**



# Dataset Analysis

## Contents

3.1	Presentation and preparation of the dataset . . . . .	34
3.1.1	Notation . . . . .	35
3.1.2	From popularity to number of views . . . . .	36
3.1.3	Metrics used to analyze the dataset . . . . .	37
3.2	Tag and view distributions . . . . .	38
3.3	Comparing the Entropy of Videos and Tags . . . . .	41
3.3.1	Video popularity and geographic distribution . . . . .	41
3.3.2	Tags and geographic distribution . . . . .	43
3.3.3	Tags as reliable markers of geographic specificity . . . . .	45
3.4	Possible mitigation of the results . . . . .	47
3.5	Summary . . . . .	48



In order to build better and more reactive decentralized UGC systems, we propose in this thesis to exploit the meta-data routinely attached to videos in such systems, with a particular focus on tags. As a first step towards this grand vision, we present in this chapter a detailed analysis of the relationship between tags and the geographic view distributions of the videos associated with these tags, on the basis of an extensive dataset of Youtube videos.

We first present our dataset in Section 3.1, along with some notations and metrics (Sections 3.1.1, and 3.1.3). We then explain how we derive the view distribution of each video from the “popularity” information provided by Youtube in Section 3.1.2. We describe a few statistics of videos and tags in our dataset (Section 3.2), before moving to an analysis of the geographic distribution of tags, and its relationship to that of videos (Section 3.3).

### 3.1 Presentation and preparation of the dataset

Our study uses a Youtube data set collected by our research group in March 2011 [40]. The seeds of the data set are the 10 most popular videos in 25 different countries, obtained through Youtube’s public API. The data set was then completed using a breadth-first snowball sampling of the graph of related videos, as reported by Youtube. For each crawled video, the data set contains, among others, the *video’s id*, its *title*, its *total number of views*, its *popularity vector* (a vector of integers representing the video’s popularity by country, more on this below), and a set of *descriptive tags* provided by the user who uploaded the video [35, 33].

The popularity vector of each video was obtained by crawling the world map which, at the time<sup>1</sup>, was provided by Youtube to indicate in which country a video was most popular. Figure 3.1, for instance, shows the world map of the video with the most views in our data set (*Justin Bieber - Baby ft. Ludacris*). Such maps were provided using Google’s Map Chart service [3] making it possible to extract for each of the 235 countries of the ISO 3166-1-alpha-2 standard an integer—from 0 to 61—representing the video’s popularity in this country (Table 3.1).

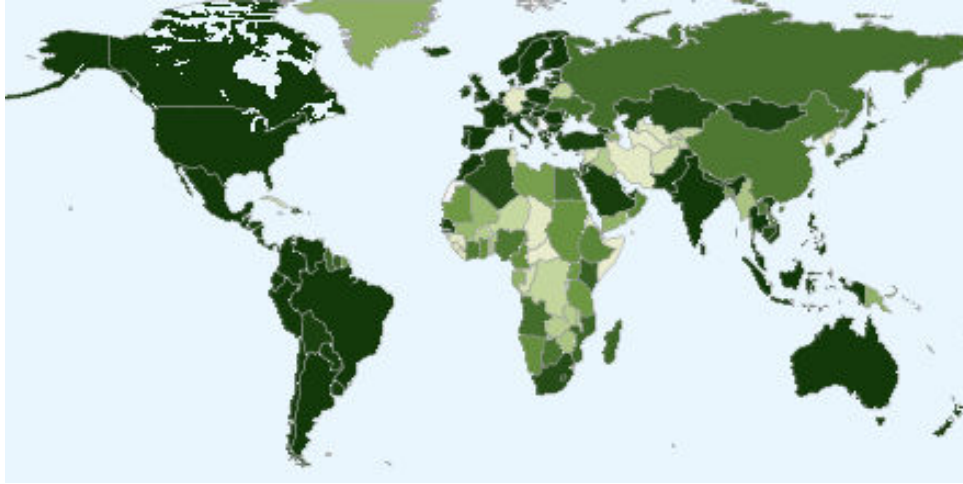
The original data set contains 1,063,844 unique videos, but not all videos have a complete set of metadata, with some lacking either tags, or their total number

---

<sup>1</sup>This information is unfortunately no longer available since YouTube changed their API and graphical user interface in September 2013, and closed access to the geographic information regarding a video’s views.

**Table 3.1: Popularity vector of the map of Fig. 3.1 (excerpt)**

US	SG	SE	RO	PT	PH	PE	NL	MY	MX	IL	...
61	61	61	61	61	61	61	61	61	61	61	...

**Figure 3.1: Popularity map of the most viewed video of our data set *Justin Bieber - Baby ft. Ludacris*, as provided by Youtube.**

of views, or their geographical distribution. Also, it was necessary to merge and reorganise the different datasets containing all the needed data, in order to forge our working version. In particular, we filtered out all videos containing no tags (6,736 videos), or with an incorrect or empty popularity vector. This filtering step resulted in a data set with 590,897 videos, associated with 705,415 unique tags, totaling 173,288,616,473 views.

### 3.1.1 Notation

For the sake of clarity, we use the following notation in the remainder of this thesis:  $\mathcal{V}$  is the set of videos in our data set. For each video  $v \in \mathcal{V}$  we use the following three pieces of information:

- $tags(v)$  is the set of tags attached to the video by the user who uploaded it. For instance, the most viewed video in our data set (Figure 3.1) is associated with the tags *Justin*, *Bieber*, *Island*, *Def*, *Jam* and *Pop*.
- $tot\_views(v)$  is the total number of views of the video;
- $pop(v)$  is popularity vector of the video as provided by Youtube.  $pop(v)[c]$  is the integer representing the popularity of  $v$  in country  $c$ .

From this information, we compute for each tag  $t$  the following sets and statistics:

- $videos(t)$  is the set of videos containing  $t$  in their tag set.

$$videos(t) = \{v \in \mathcal{V} \mid t \in tags(v)\} = tags^{-1}(t)$$

- $freq(t)$  is the number of occurrences of  $t$ , i.e.

$$freq(t) = |videos(t)|$$

- $tot\_views(t)$  is the total number of views associated with  $t$ , i.e. the aggregated number of views of the videos containing  $t$ .

$$tot\_views(t) = \sum_{v \in videos(t)} tot\_views(v)$$

### 3.1.2 From popularity to number of views

The exact meaning of the popularity vector  $\mathbf{pop}(v)$  is not documented by Youtube. This vector is however unlikely to capture the proportion of a video's views originating from individual countries: applied to Table 3.1, this assumption would imply that the video *Justin Bieber - Baby ft. Ludacris* has been viewed as many times in the USA (US, population 318.5M) as in Singapore (SG, population 5.4M).

Instead, taking cue from *Google Trends* [4], one of the analytics services provided by Youtube's parent company Google, we consider a video's popularity vector to represent the *intensity* of this video in individual countries, i.e. a number proportional to the share of this video's views in this country's Youtube traffic:

$$\mathbf{pop}(v)[c] = \frac{\mathbf{views}(v)[c]}{\mathbf{ytube}[c]} \times K(v) \quad (3.1)$$

where  $\mathbf{views}(v)[c]$  is the number of views of  $v$  in country  $c$ ,  $\mathbf{ytube}[c]$  is the total number of Youtube views in country  $c$ , and  $K(v)$  is a normalization factor, dependent of each video, to scale values in the range [0 – 61].

Neither  $\mathbf{ytube}[c]$  nor  $K(v)$  are available to us. To estimate both, we use the distribution of Youtube traffic provided by Alexa Internet Inc. [5] on July 2014, an authoritative source of internet traffic, and statistics on internet usage and

users per country provided by the International Telecommunication Union [6] to approximate the distribution of Youtube views per country:

$$\mathbf{ytube}[c] = \mathbf{p}_{yt}[c] \times T_{yt} \simeq \widehat{\mathbf{p}}_{yt}[c] \times T_{yt} \quad (3.2)$$

where  $\mathbf{p}_{yt}[c]$  is the proportion of Youtube view in country  $c$  at the time our data set was collected,  $T_{yt}$  is the total number of Youtube views at the same time, and  $\widehat{\mathbf{p}}_{yt}[c]$  is the Youtube traffic estimated by Alexa for country  $c$ .

We also use the fact that we know the total number of views of each video in our data set:

$$tot\_views(v) = \sum_{c \in World} \mathbf{views}(v)[c] \quad (3.3)$$

Injecting (3.2) in (3.1), and (3.1) in (3.3) eliminates  $\mathbf{ytube}[c]$ ,  $K(v)$  and  $T_{yt}$ , and yields the following formula:

$$\mathbf{views}(v)[c] \simeq \frac{\widehat{\mathbf{p}}_{yt}[c] \times \mathbf{pop}(v)[c]}{\sum_{\gamma \in World} (\widehat{\mathbf{p}}_{yt}[\gamma] \times \mathbf{pop}(v)[\gamma])} \times tot\_views(v) \quad (3.4)$$

This formula provides us with the geographic distribution of the views of each videos. For each tag  $t$ , we derive from these distributions the number of views associated with  $t$  in country  $c$  (noted  $\mathbf{views}(t)[c]$ ) using the same proportional mechanism as above, yielding the aggregated number of views in country  $c$  of the videos containing  $t$  as tag.

$$\mathbf{views}(t)[c] = \sum_{v \in videos(t)} \mathbf{views}(v)[c] \quad (3.5)$$

### 3.1.3 Metrics used to analyze the dataset

In this analysis, we are particularly interested in capturing a tag's geographic spread (resp. concentration), and in contrasting this spread to the videos associated with this tag. To this aim, we use Shannon's entropy  $H(t)$  on the view distribution of a tag  $t$  (resp. video  $v$ ) among countries:

$$H(x) = - \sum_{c \in World} \mathbf{p}_{geo}(x)[c] \times \log_2(\mathbf{p}_{geo}(x)[c]) \quad (3.6)$$

where  $x$  is either a video or a tag, and  $\mathbf{p}_{geo}(x)[c]$  represents the proportion of views of this video or tag in country  $c$ :

$$\mathbf{p}_{geo}(x)[c] = \frac{\mathbf{views}(x)[c]}{tot\_views(x)}$$

**Table 3.2: The 10 most frequent tags**      **Table 3.3: The 10 most viewed tags (worldwide)**

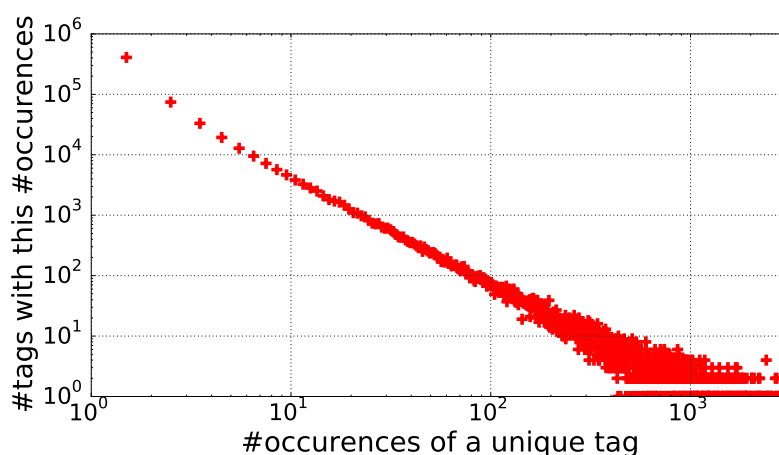
tag	#occur	#views	average #views	tag	#occur	#views	average #views
the	30686	13,157,705,562	428,785	funny	21645	13,550,709,569	626,043
video	27239	12,898,383,171	473,526	pop	7877	13,318,507,233	1,690,809
music	23128	12,640,171,764	546,531	the	30686	13,157,705,562	428,785
2010	22014	3,349,620,292	152,158	video	27239	12,898,383,171	473,526
funny	21645	13,550,709,569	626,043	music	23128	12,640,171,764	546,531
of	19820	5,940,302,641	299,712	of	19820	5,940,302,641	299,712
new	17943	5,293,119,879	294,996	records	2478	5,920,162,042	2,389,088
2011	14572	756,842,996	51,938	hip	5085	5,615,505,842	1,104,327
live	11614	3,196,117,558	275,195	hop	5047	5,615,431,517	1,112,627
de	11314	2,726,151,223	240,953	comedy	9039	5,603,654,002	619,941

A high entropy means a tag (or video) tends to be spread uniformly among many countries. By contrast, a low entropy denotes a tag (video) whose views are concentrated in a few countries. For instance, the video with the highest number of views in our data set, *Justin Bieber - Baby ft. Ludacris* shown in Figure 3.1, has an entropy of 5.06. This value is close to the highest possible value of  $\log_2(235) = 7.87$ , which would correspond to a video equally distributed among the 235 countries tracked by Youtube. By contrast, the lowest possible entropy value is  $\log_2(1) = 0$ , corresponding to a tag (video) whose views originate from one single country.

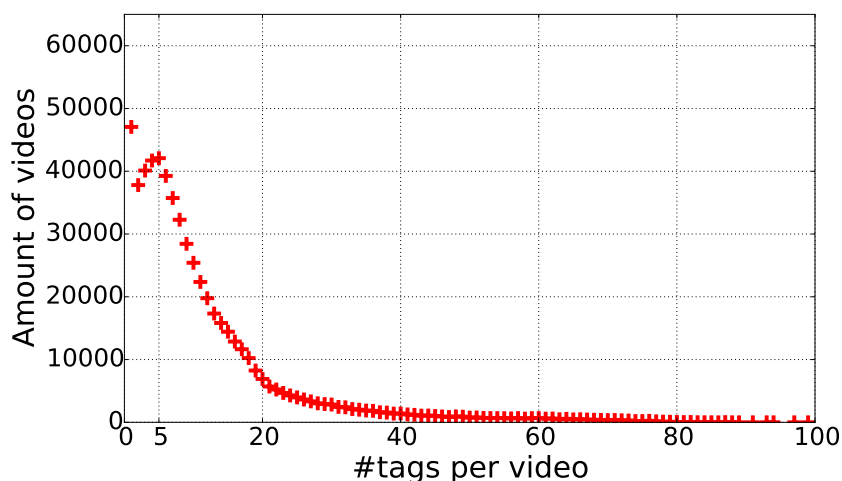
## 3.2 Tag and view distributions

Our data set contains 7,717,815 tag occurrences, for an average of 11.18 tags per video, and a total of 705,415 unique tags. This large number of tags, in line with earlier findings [33], can be explained by the presence of compound tags (e.g. “*korean pop*” is different from “*korean*” “*pop*”, which counts as two tags), spelling mistakes (“*music*” or “*music\_*” instead of “*music*”), and the use of multiple languages. The frequency distribution of individual tags (Figure 3.2) shows a typical power-law, which is commonly found in natural languages and folksonomies. About 462,549 tags (66%) only appear once.

Tables 3.2 and 3.3 show, respectively, the 10 most frequent tags and the top 10 tags with the most views. Most tags describe content (*video*, *funny*) but some consist of syntax tools (*the*, *of*). The latter probably result from the former usage of spaces to separate tags (Youtube now uses commas), which caused compound terms such as *the\_rock* to be parsed into two tags (*the* and *rock*). The tables also



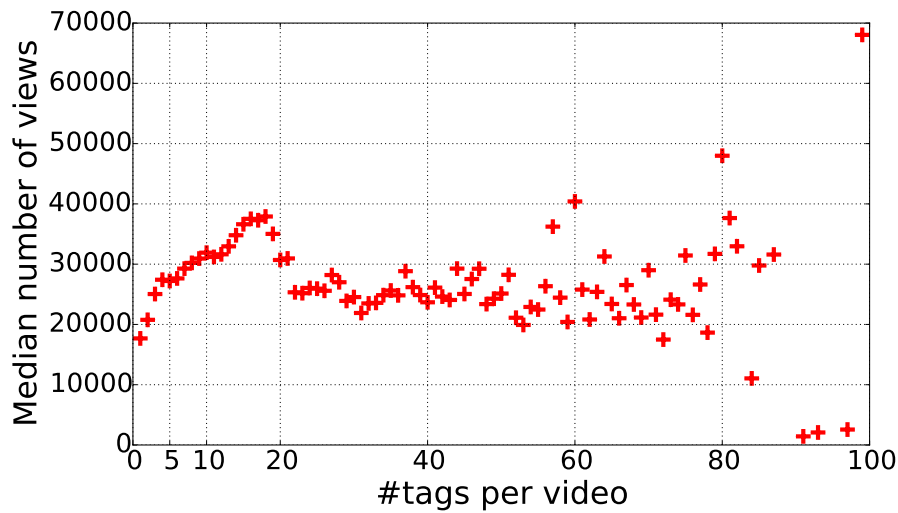
**Figure 3.2:** The frequency distribution of tags follow a power law of the shape  $y = K \times x^{-\alpha}$ , as often observed in folksonomies and natural languages.



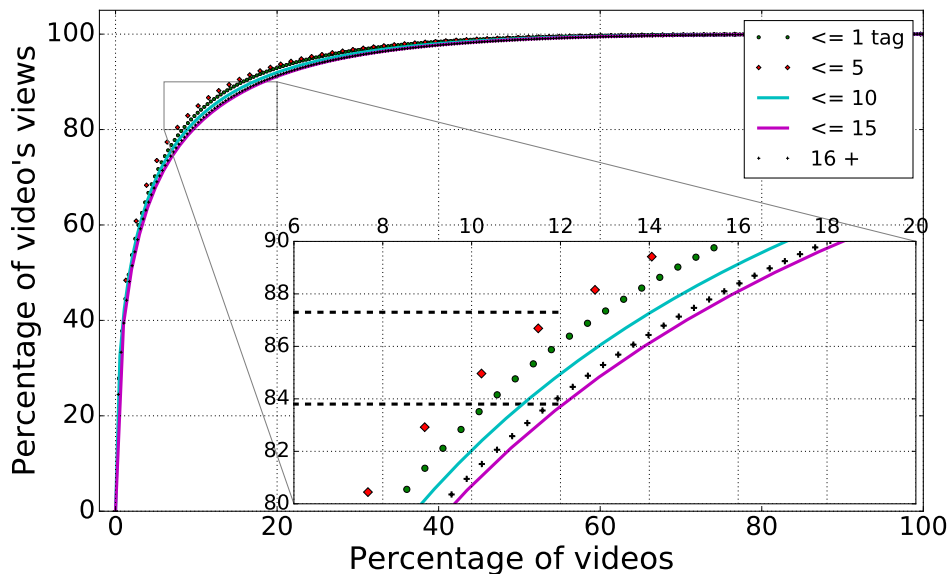
**Figure 3.3:** Tags are widely used to describe videos, with 50% of videos showing a least 11 tags.

show that the most viewed tags are not necessarily the most frequent. For example, *pop*, the second most viewed tag (Table 3.2), only occurs 7877 times. The corresponding videos predominantly belong to the “Music” category, with a high average number of views per individual video (1,690,809 views, 2.7 times more than those of videos containing the tag *funny*). The same comment applies to related tags such as *hip* and *records*.

As mentioned, videos have relatively rich tag descriptions (Figure 3.3) with 11.18 tags on average. One reason may be that users have an incentive to tag their videos to attract more views. However, and perhaps surprisingly, there seems to be only a weak link between the number of tags of a video and this video’s



**Figure 3.4:** Median number of views for the videos embedding a given number of tags. Views and size of the tag set seem only weakly correlated, with a clear growing trend limited to videos with less than 18 tags. The high range results are not relevant due to the small number of videos having that number of tags, as show in Figure 3.3



**Figure 3.5:** Cumulative distributions for video's views, by number of tags. The number of tags have little to none influence on the views distributions.

viewership (Figure 3.4). The median number of views of a video increases with up to 18 tags. But this relationship collapses beyond this value. For instance, the most tagged video in our data set possesses 102 tags, but only 1,220,496 views, which pales in comparison to the most viewed video—471,208,788 views for only 6 tags. This weak or absent correlation between number of tags and number of

views is also apparent in Figure 3.5, which shows the proportion of aggregated views, as a function of the proportion of videos categorized in different ranges of tag numbers. Figure 3.3 also allows us to note that more than 400,000 unique tags occurs only once in that dataset. The most plausible reason is a bad syntax for these tags, as explained earlier.

In the following, in order to avoid artefacts caused by videos with very low numbers of views, we only consider videos with at least 1000 views. We also limit our discussion to iso-latin1 tags (91.03% of all tag occurrences). This yields our final dataset, containing 591,409 videos, 628,101 unique tags, and an aggregated total of 173,248,620,343 views.

### 3.3 Comparing the Entropy of Videos and Tags

To understand how tags can provide information to drive the storage of videos, we now analyse the geographic distributions of videos and tags in our data set.

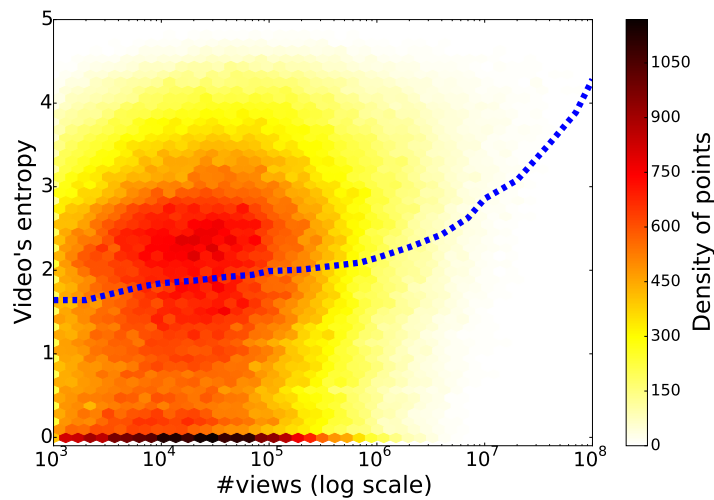
#### 3.3.1 Video popularity and geographic distribution

We first start by considering videos, and analysing the relationship between their popularity and their geographic distribution. Figure 3.6 depicts this relationship in the form of a heat map. The  $x$  axis represents the popularity of videos in terms of their number of views, the  $y$  axis measures the geographical distribution in terms of entropy, while colours indicate the density of videos with the corresponding entropy-popularity values.

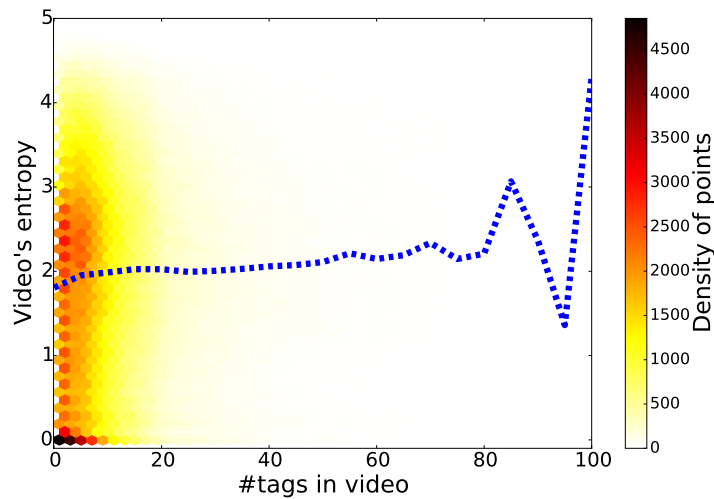
As pointed out in earlier work [40], the views of popular videos, in particular those with more than  $10^7$  views, tend to be widely distributed, with average entropy values between 3 and 4. These high entropy values mean that these videos need to be easily accessible from all over the world, which reduces the interest in predicting their geographical distribution [20]. However, the plot also shows that these popular videos constitute a minority. Most of the data points in the heat map represent videos with less than  $10^6$  views. For these videos, the average entropy remains around the value of 2, with a few high density points around entropy values of 2.5, 1.5 and 0.

These numbers show that a significant fraction of videos are geographically concentrated. For example, videos with an entropy below 1.5 constitute 40% of the data set, with a mean number of views of 155,520, a mean number of tags of 9





**Figure 3.6: Heatmap of each video's entropy vs. its number of views. Mean shown as a dashed line.**



**Figure 3.7: Heatmap of each video's entropy vs. its number of embedded tags. Mean shown as a dashed line.**

(vs. 11.18 for the whole data set), and a mean entropy of 0.707. To get a feel of the meaning of these numbers, we observe that an entropy of 1.5 could, for example, correspond to a video that is present and uniformly distributed in only 4 countries. In general, such a low value corresponds to videos that are geographically concentrated and thus that constitute perfect candidates for proactive placement strategies. The observations in Figure 3.7 add up to the statement that a significant fraction of videos are concentrated in restricted areas. The highlight of two distinct high-density zones, with one demonstrating a cluster of widely spread video audiences corresponding to popular content observed before, and a second with a much higher density, with very low entropy scores. This last graphical observation allows us to eliminate the possibility of existence of a proportion-

**Table 3.4: The most viewed tags for various countries**

country	tag	total views
United-States	funny	7,907,521,226
Germany	music	557,388,816
France	pop	536,096,206
Canada	funny	484,758,340
Australia	funny	236,812,186

**Table 3.5: Top 3 Videos (views) containing *pop***

title	#views	%
<i>Justin Bieber - Baby ft. Ludacris</i>	471,208,788	3.54%
<i>Lady Gaga - Bad Romance</i>	348,924,582	2.62%
<i>Shakira - Waka Waka ...</i>	306,374,501	2.30%
<i>total for top 3</i>	<i>1,126,507,871</i>	<i>8.46%</i>

nal relation between the geographical spread of views and the number of tags attached to that content.

### 3.3.2 Tags and geographic distribution

In a second move, we argue that tags can contribute to place these geographically concentrated videos close to their viewers. To verify this hypothesis, we start by analysing the most popular tags. Table 3.4 shows that the most viewed tag in each of five western countries (France, Germany, Canada, Australia, and USA) is *music* (entropy of 3.80), *pop* (entropy of 4.27) or *funny* (entropy of 3.03).

Based on this example, one might wonder if the popularity of tags correlates with that of the corresponding videos. But this is not necessarily the case. For example, the top three videos with the tag *pop* (Table 3.5) also turn out to be the most viewed in the entire data set. However, other tags, like *funny*, appear in a large number of possibly much less popular videos. To assess the potential of tags for predicting the consumption of videos we therefore seek for a correlation between their entropy values.

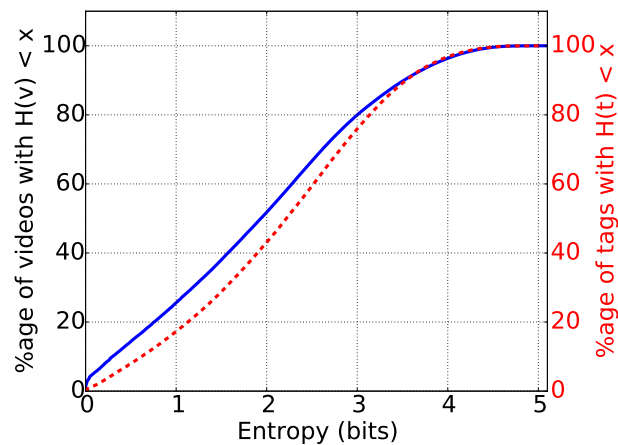
Figure 3.8 compares the cumulative distribution function (CDF) of the entropy of videos (solid line) with that of tags (dashes) in our data set. The two curves exhibit very similar trends: entropy values tend to be evenly spread for values below 3, which correspond to roughly 80% of all videos and tags. Table 3.6 complements this information by showing the tags with the highest and those with the lowest entropy.

**Table 3.6: The 5 tags with the most (above) resp. least (below) entropy (for #occurrences > 100)**

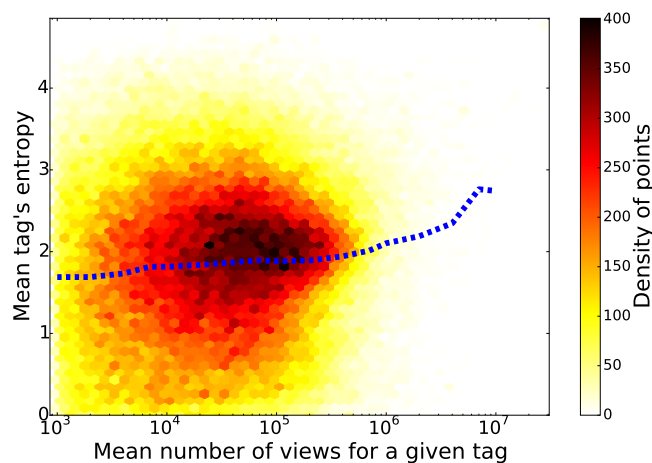
tag	H(t)	#occurs	#views	average views
recovery	4.90	230	557,870,332	2,425,523
dominic	4.87	103	338,555,233	3,286,944
fifa	4.83	2722	690,092,931	253,524
passat	4.79	142	41,809,394	294,432
afraid	4.78	131	244,659,961	1,867,633

tag	H(t)	#occurs	#views	average views
piologo	0.04	101	3,985,341	39,458
mundo canibal	0.06	134	4,147,866	30,954
kvarteret	0.10	102	7,313,481	71,700
skatan	0.11	106	7,741,235	73,030
partoba	0.18	272	7,183,083	26,408



**Figure 3.8: CDF of videos (solid line) and tags (dashes) versus entropy**



**Figure 3.9: Heatmap of the mean views for every occurrences of a given tag, versus the mean entropy of every occurrences of that tag. Mean showed as a dashed line.**

Table 3.7: Top 5 countries (views) for *bollywood*

country	#views	%age
India	200,956,055	39.8%
United-States	124,461,447	24.7%
United-Kingdom	29,506,586	5.8%
Pakistan	25,218,518	5.0%
Germany	12,842,983	2.5%

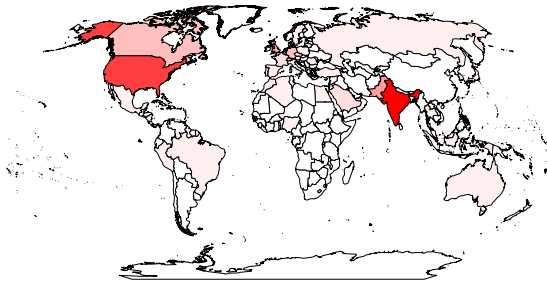


Figure 3.10: Videos associated with the tag 'bollywood' tend to be viewed mainly in India, USA and UK.

Table 3.8: 5 top countries (views) for *favela*

country	#views	%age
Brazil	19,834,633	47.9%
United-States	14,468,608	34.9%
United-Kingdom	1,701,496	4.1%
Canada	785,725	1.9%
Mexico	639,375	1.5%

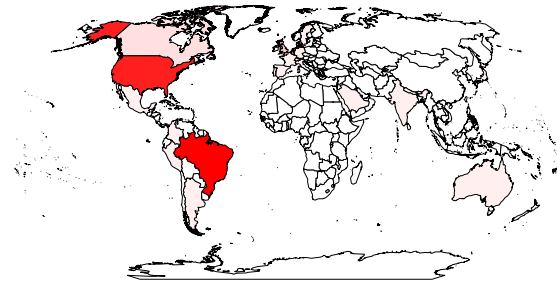


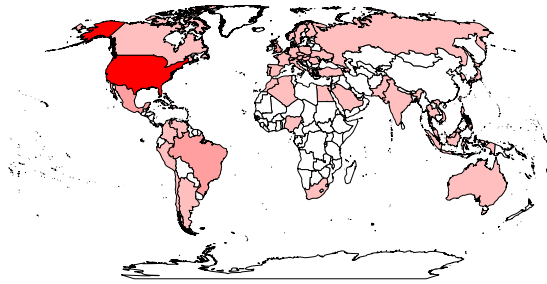
Figure 3.11: Videos associated with the tag 'favela' are mostly viewed in Brazil

### 3.3.3 Tags as reliable markers of geographic specificity

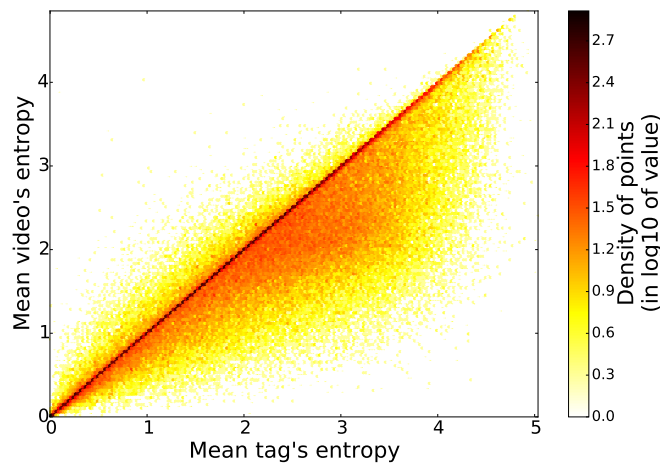
Figure 3.9 depicts the relationship between the entropy and the popularity of tags in the form of a heat map. As for videos, popular tags constitute a minority: most tags have entropy values around 2, and an average of 100,000 views. We provide two examples of such specific tags in Table 3.7 and Figure 3.10, and in Table 3.8 and Figure 3.11. The two tables and figures show the top-5 viewing countries and the viewership distribution for tags *bollywood* (entropy of 3.24) and *favela* (entropy of 2.22). In the figures, a higher color saturation indicates a higher proportion of views for the corresponding country. The views of *bollywood* mostly occur in India and the United-States (64.5%), as expected for cultural and language reasons, with three additional countries with important South Asian minorities accounting for another 11.3%. The views of *favela* are even more concentrated with Brazil responsible for almost 48% of all views, followed by the United-States with 34.9%. On the other side, when we visualize the distribution of views for very popular tags such as *pop* that are widely spread and consumed, with an entropy of 4.27 in this case, we observe different characteristics. Table 3.9 demonstrate a very high interest in one country, with the United-States counting for 35.2% of the views, but then the following top countries in shares

**Table 3.9: Top 5 countries (by views) for *pop***

country	#views	%age
United-States	4,700,159,350	35.2%
United-Kingdom	759,449,112	5.7%
Brazil	751,342,295	5.6%
Mexico	603,876,310	4.5%
India	586,339,771	4.4%



**Figure 3.12: Videos associated with the tag 'pop' tend to be uniformly distributed over the globe, taking into account the number of YouTube users in a country.**

**Figure 3.13: Tag entropy versus video entropy**

of consumption have little to no differences between them. The scattering of consumption presented in Figure 3.12 appears almost equal within the countries having a rather generally democratized access to Internet. This tag is a marker of a very wide interest towards contemporary successful and famous artists, producing pop-music. The predominance of consumption from the United-States could be explained by the fact that the three most viewed videos of our dataset are incorporating the tag *pop* are produced by United-States artists.

These graphical representations push forward that some tags indicates clearly that their attached contents are consumed in very restricted areas. These cat-

egory of tags are not among the most popular but are the most numerous. The figures supports for the most part this claim. An extend of that observation is to note that these figures suggest that caching or storing copies of videos containing these tag in the respective top countries would significantly benefit UGC video systems.

This claim is backed up by the correlation of entropy observed in this last figure. Figure 3.13 plots the mean entropy of each unique tag versus the mean entropy of all the videos this tag appears in. The plot exhibits mainly a linear shape, indicating that for most pair (tag, video), the tag's entropy and the video's entropy are strongly correlated.

A thin *spread-belt* surrounds the relation, with an almost constant width, pointing in advance towards the level of reliability of the tags as geographic consumption markers of a content. The overall relation is strong with the higher density concentration being found on the diagonal. This strong link reinforces our conjecture that tags can predict the geographic distribution of the associated videos.

### 3.4 Possible mitigation of the results

We acknowledge that the entire validity of our work can be discussed in various ways. First, we understand that we rely on a large data set, but not large enough to be entirely meaningful with respect to the current size of YouTube. It is still possible that our observations may be an artefact of the particular choice of videos that constitute our data set. The changes in the YouTube API that happened in 2013 prevented us from crawling a new dataset or from enhancing the one we presented.

A second point that may raise some discussion lies in the way we used the geographic vector provided in the metadata to obtain the proportion of views per country for each video. To the best of our knowledge, the only way to make a more informed choice would require having access to YouTube's inner functions.

Another limitation of our work derives from the fact that our dataset has a country-level granularity. Previous studies, that we already cited in Chapter 2 demonstrate that consumption trends tend to be homogeneous inside a country. Nonetheless, we recognise that having a granularity at the level of metropolitan areas could improve the proximity of content placement, on a geographic and network scale.

We recognise that the above points can have a direct influence on our results. For this reason, we took some proactive measures, such as testing Mignon, the contribution in Chapter 5 on two different datasets, to further support our conclusions.

### 3.5 Summary

Our analysis of a large dataset of Youtube videos has confirmed that user-defined tags were widely used to describe videos, and that an important proportion of tags and videos showed distribution concentrated in a few geographic areas. Furthermore, it appears that a video with a low entropy tends to be associated with tags that also have a low entropy, and reciprocally. This last point suggests that tags can be used to estimate whether the views of a new video will be widely distributed or concentrated in only a few countries, and in this second case, what these countries might be.

**This raises however two questions: how might such a tag-based prediction of video-views work? Does it holds the potential to help construct decentralized UGC systems?** These are the two questions we investigate in the next chapter.

## Potential of the tags

### Contents

4.1	Predicting views shares from tags . . . . .	50
4.1.1	General approach . . . . .	50
4.1.2	Baseline . . . . .	51
4.1.3	Evaluation and metric . . . . .	51
4.1.4	Results . . . . .	53
4.2	Potential of tags for proactive video placement . . . . .	55
4.2.1	System model . . . . .	56
4.2.2	Placement mechanism . . . . .	57
4.2.3	Experiment, metrics, baseline . . . . .	57
4.2.4	Results . . . . .	58
4.3	Summary . . . . .	60



In the previous chapter, we analysed a rich dataset from YouTube collected by our team in 2011, in order to highlight the strong correlation relations between a video’s views geographic distribution and the ones of the associated tags. This fundamental part allowed us to have an inkling of the predictive possibilities that tags can give us on the geographic consumption of contents. In this chapter, we go one step further and we establish the potential of tags to rely on, in order to predict the geographic consumption pattern of a video content and their ability to be used as a core element of a proactive video placement system.

In the rest of the chapter, we first propose in Section 4.1 a simple approach to predict view distributions from the tags attached to a video, and evaluate this approach on the datasets presented in Chapter 3. We then explore how the predictive power of tags could be used to improve the cache behavior of a decentralized UGC system (Section 4.2), and evaluate our proposal on the same dataset. Section 4.3 concludes the chapter with a brief summary.

## 4.1 Predicting views shares from tags

The problem we wish to solve is the following: **Given a new video  $v$ , and  $v$ ’s tags, we wish to predict the geographic distribution of  $v$ ’s future views, knowing the distribution of past videos with the same tags.** Because we eventually would like to distribute this prediction in a decentralized setting, it should remain as simple as possible, while still providing suitable results.

To compute this prediction, we use a basic additive prediction technique that exploits the tags associated with videos (Section 4.1.1), and compare it with a baseline prediction mechanism (Section 4.1.2). To evaluate both approaches, we use a cross-validation technique. We split the dataset into a testing set  $\mathcal{V}_{\text{test}}$  and a training set  $\mathcal{V}_{\text{train}}$ . We then process the information (views and tags) in the training set, and use it to guess the view distributions of the videos in the testing set.

### 4.1.1 General approach

When a new video  $v$  is uploaded, we predict the geographic distribution of  $v$ ’s views  $\widehat{\mathbf{p}}_{\text{geo}}(v)$  as the average of the geographic distribution of  $v$ ’s tags in the set of videos already known to the system  $\mathcal{V}$ : For a video  $v \in \mathcal{V}_{\text{test}}$  associated with a set of tags  $\text{tags}(v)$ , we predict the geographic distribution of  $v$ ’s views  $\widehat{\mathbf{p}}_{\text{geo}}(v)$  as the

**Table 4.1: YouTube traffic share according to Alexa**

country	share	country	share
United-States	19.0%	United-Kingdom	3.2%
India	8.6%	Mexico	3.0%
Japan	4.7%	Germany	3.0%
Russia	4.1%	France	2.5%
Brazil	3.8%	Spain	2.3%

average of the geographic distribution of  $v$ 's tags in the training set  $\mathcal{V}_{\text{train}}$ :

$$\widehat{\mathbf{p}}_{\text{geo}}(v) = \mathbb{E}_{t \in \text{tags}(v)} \left( \mathbf{p}_{\text{geo}}^{\mathcal{V}}(t) \right) \quad (4.1)$$

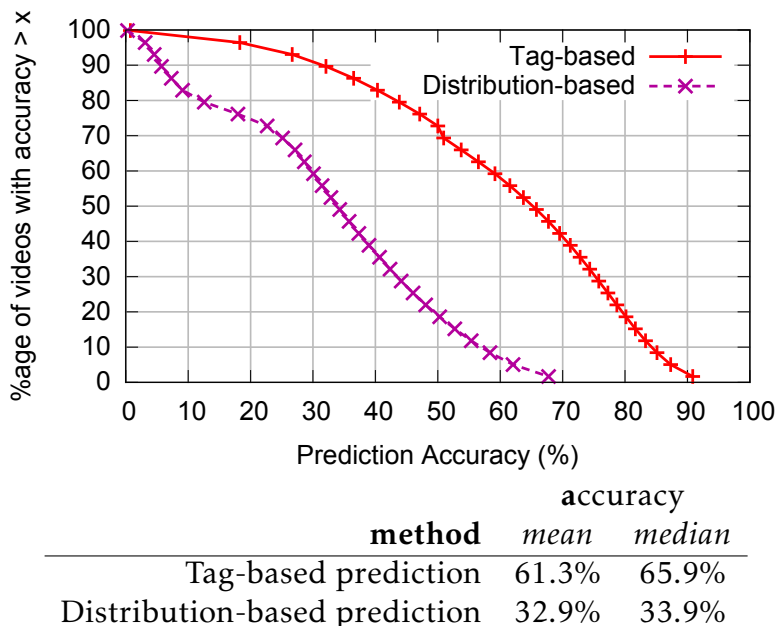
where  $\mathbf{p}_{\text{geo}}^{\mathcal{V}}(t)$  is the geographic distribution vector of tag  $t$  in the dataset.  $\mathcal{V}$ . Our aim is for  $\widehat{\mathbf{p}}_{\text{geo}}(v)$  to be as close as possible to  $\mathbf{p}_{\text{geo}}(v)$ ,  $v$ 's actual view distribution vector.

### 4.1.2 Baseline

As a baseline prediction, we use the average distribution of global YouTube views, estimated from the YouTube network traffic reported by Alexa Internet Inc. [1]. Table 4.1 lists the 10 countries having the biggest shares of traffic. We use this data as viewing probabilities. With reference to Table 4.1, a content has 19% of probabilities to be viewed in the USA, 8.6% chances to be viewed in India, and so on. Alexa only provides for YouTube the top 40 countries generating the most traffic, totalling 85.2% of global YouTube network usage. We apportion the remaining portion of 14.8% to the 217 countries not covered by Alexa proportionally to their share of internet users, as reported by the International Telecommunication Union [6]. This process yields the same baseline view prediction for all videos, that we compare against the results returned by (4.1).

### 4.1.3 Evaluation and metric

We evaluate the tag-based prediction strategy and compare it with the baseline using a cross-validation method. We divide our dataset into two equal parts: a training set  $\mathcal{V}_{\text{train}}$  and a testing set  $\mathcal{V}_{\text{test}}$ . By ranking the videos by number of views, and attributing each odd rank order to the  $\mathcal{V}_{\text{train}}$  and each even rank order to  $\mathcal{V}_{\text{test}}$  we obtain a training set and a testing set containing 295,449 ( $\pm 1$ ) videos each, and a very close number of views for both 86,624,310,171 ( $\pm 20,000,000$ ).



**Figure 4.1: CDF of prediction accuracy (top) and mean and median (bottom) for the tag-based and distribution-based approaches for view prediction (higher is better). Tags clearly yield better predictions over a simple average distribution vector.**

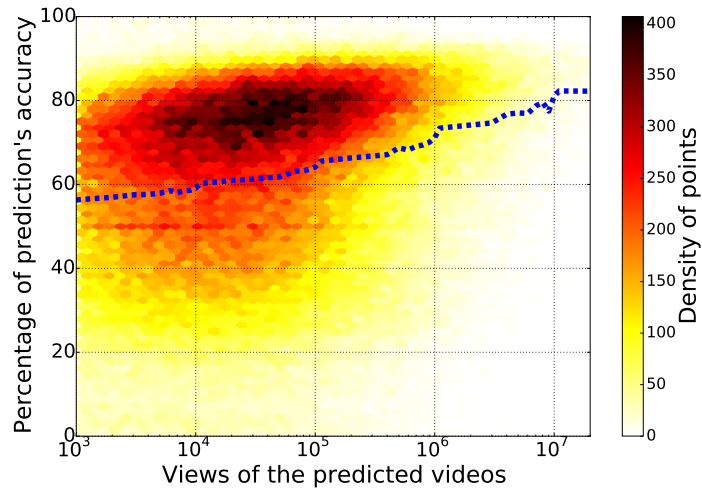
We then use (4.1) to predict the view distribution of each video  $v$  in  $\mathcal{V}_{\text{test}}$  from the tag distribution extracted from  $\mathcal{V}_{\text{train}}$  (which plays the role of known videos in the formula).

To evaluate the accuracy between a prediction  $\widehat{\mathbf{p}}_{\text{geo}}(v)$  and the actual geographic distribution of a video  $\mathbf{p}_{\text{geo}}(v)$  we compute the proportion of views correctly placed by the prediction (what we term the prediction’s *accuracy*):

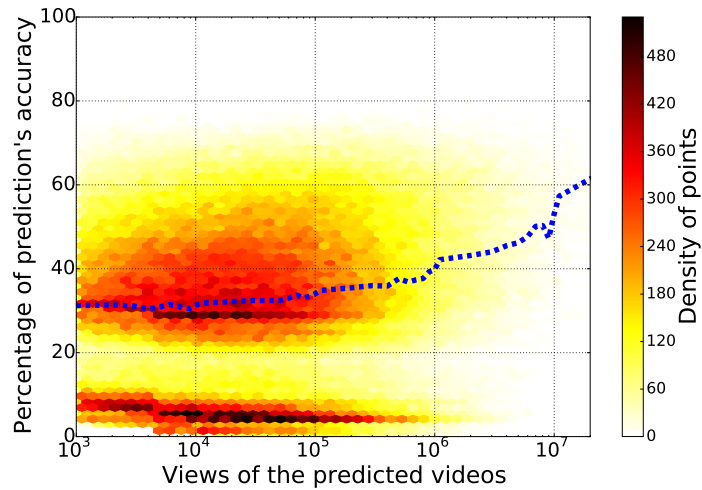
$$\mathbf{p}_{\text{correct}}(v) = 1 - \frac{1}{2} \times \sum_{c \in \text{World}} \left| \mathbf{p}_{\text{geo}}(v)[c] - \widehat{\mathbf{p}}_{\text{geo}}(v)[c] \right| \quad (4.2)$$

where  $\mathbf{p}_{\text{geo}}(v)$  is the actual geographic distribution of video  $v$ , and the division by 2 normalizes the result. An accuracy of 1 means that the prediction and the actual distributions match (no misplaced views); a value of 0 instead indicates there is no overlap in terms of countries between the predicted and actual views (all views were misplaced).

In the following, we present our results and show how prediction accuracy can be influenced by parameters such as the number of views, the number of tags, or the entropy of tags and videos.



(a) Tag-based prediction

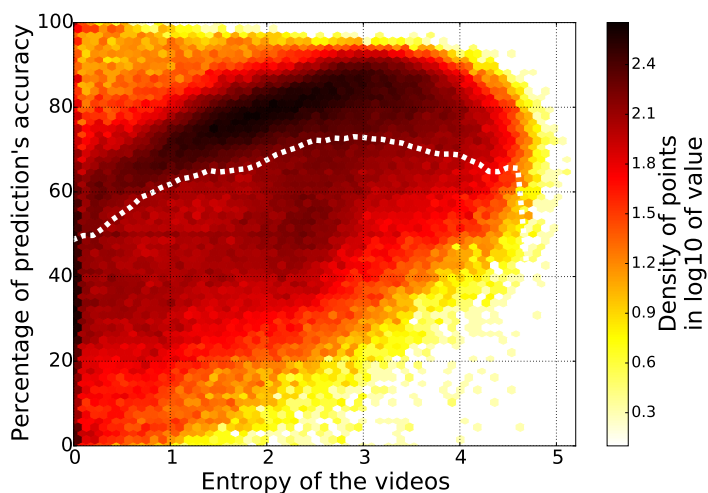


(b) Distribution-based prediction

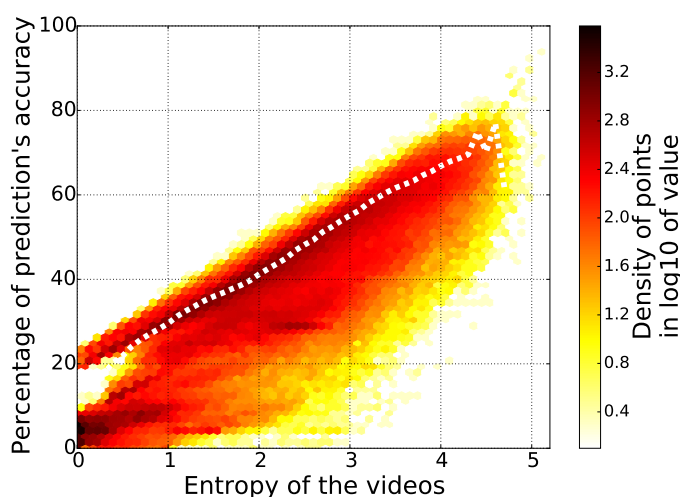
**Figure 4.2: Prediction accuracy vs video views.** The dashed lines show the average accuracy. The tag-based approach outperforms the baseline across the range of video views.

#### 4.1.4 Results

We start presenting our results by comparing the distributions of prediction accuracy for our tag-based approach and for the baseline view prediction. Figure 4.1 plots the cumulative distribution of prediction accuracy obtained by our approach (*Tag-based prediction*) and by the baseline (*Distribution-based prediction*) with the corresponding mean and median values indicated below. Our approach clearly outperforms the baseline, yielding a median accuracy (65.9%) that is almost twice that of its competitor (33.9%). We note that our *Tag-based prediction* leads us to obtain more than 50% of accuracy for 73% of videos compared to



(a) Tag-based prediction



(b) Distribution-based prediction

**Figure 4.3: Prediction accuracy vs video entropy.** The dashed lines show the average accuracy. The benefit of tags is particularly strong for low entropy values.

only 19% for the *Distribution-based prediction*. This confirms that tags hold the promise of predicting the geographic distribution of UGC videos.

Figures 4.2 and 4.3 delve deeper into the results and show the effect of the number of views and of the entropy of a video (see Chapter 3), respectively, on the accuracy of prediction for both approaches. The heat maps show the distribution of individual videos, while the dashed lines indicate the average accuracy obtained for a given number of views, resp. entropy.

Figure 4.2 indicates that tag-based prediction significantly outperforms the

baseline regardless of a video's popularity with the absolute difference between the accuracy values of the two approaches remaining at about 30% over all popularity values. Both plots further show a weak positive correlation between the number of views of a video and its accuracy. This correlation probably stems from the link between popularity and entropy. Highly popular videos tend to be scattered all over the world (high entropy), and are therefore easier to predict, as when the interest towards a content is shared by everyone all around the world, the consumption tends to copy the same pattern as the share of internet users per country.

By contrast, Figure 4.3 shows that tag-based prediction works best for video with an average to medium-high entropy (between 2 and 3, accuracy above 70%), with lower results both for both highly concentrated and widely distributed videos (corresponding to low resp. high entropy values). This behaviour is in stark contrast to that of the baseline, whose performance is directly linked to that of entropy, indicating that the predicting value of tags is particularly interesting for videos with low to medium entropy, which tend to diverge from the average behaviour. It can be explained by the fact that tags with a very high entropy are not specifically related to highly viewed content. This tags tend to belongs to common description as seen before in Table 3.6, *recovery*, *afraid* and *fifa* belongs to content who found interested consumers in almost every country in the world, but with a weaker fondness of the population than for popular pop singers. The interest for them is so high among any population that the consumption scheme of their contents tends to match the connectivity pattern. That is why tags belonging to content with low or medium view scores but high geographic spread benefits from a less precise prediction ability.

## 4.2 Potential of tags for proactive video placement

The results of Section 4.1 show that the tags attached to videos can be used to predict where individual videos will be most viewed with a reasonable accuracy, using a very direct and simple approximation method.

Building on this result, we now explore whether tags can help design better UGC systems by determining where to place new videos. This ability will become increasingly important as more and more applications manage short-lived content preferentially in locations closed to where they are likely to be most viewed.

### 4.2.1 System model

Our scenario considers a company that must decide where to store the primary copies of a set of new videos  $\mathcal{V}_{\text{new}}$  on its global storage infrastructure (i.e. these copies form the reference storage of the UGC service, in contrast to caching copies, which might be evicted), using tag information extracted from videos already served by the service  $\mathcal{V}_{\text{known}}$ .

In the following we focus on the placement and storage of the videos in  $\mathcal{V}_{\text{new}}$ , since those are the video for which the company has no viewing information, and where predictions are likely to be most useful.

We need some illustrations. Can you reuse some of those you drew for your presentations? (and refer to them in the text.)

In terms of infrastructure, we consider an extreme case, in which each country has some storage capacity available for new videos (a datacenter, or share of datacenter for small countries). We assume the system's overall available capacity ( $\mathcal{S}_{\text{world}}$ ) is able to store  $R$  copies of each new video. For the rest of the chapter, we will consider the replica factor  $R$  as  $R = 3$ , being a typical value for  $R$  used in cloud storage systems (e.g. GFS, HFS). For simplicity's sake, we also consider that all videos has the same size (an obvious simplification), and measure our storage capacity in number of videos.

We assume that the service's revenues, and hence its investment, will be roughly proportional to the number of views in one country, reflecting the level of activity. We therefore set the storage capacity  $\mathcal{S}_c$  of each country  $c$  proportional to the country's view shares:

$$\mathcal{S}_c = \mathcal{S}_{\text{world}} \times \mathbf{p}_{\text{global}}[c]$$

where  $\mathbf{p}_{\text{global}}[c]$  is the proportion of views in country  $c$  with  $\mathcal{S}_{\text{world}}$  representing the aggregated storage capacity of each country :

$$\mathcal{S}_{\text{world}} = \sum_{c \in \text{World}} \mathcal{S}_c = R \times |\mathcal{V}_{\text{new}}|$$

and  $|\mathcal{V}_{\text{new}}|$  representing the number of videos we want to store and serve, present in our testing set. UGC providers typically rely on multiple layers of caches (within browsers, at Internet Points-of-Presence, within datacenters), in addition to their primary storage system [39]. In our model, we aggregate all these caches in one single layer located within each country, set to an LRU eviction policy. We set the capacity of this caching layer to 10% of each country's primary storage.

This value is relatively low on purpose in order to better analyze the effect of tags on the system.

### 4.2.2 Placement mechanism

Our goal consists in finding a good placement for the copies of the videos in  $\mathcal{V}_{\text{new}}$  by using the tag information contained in  $\mathcal{V}_{\text{train}}$ . A good placement is one that maximizes the number of video requests that are served from a copy stored in the country’s local storage infrastructure.

To demonstrate the potential of tags to help organize the video storage of a UGC service, we propose to use the following simplistic approach.

We place each new video  $v$  according to an estimation of its per-country viewing vector  $(\widehat{\mathbf{views}}(v)[c])_{c \in \text{World}}$ . This estimation uses the geographic distribution of tags observed in the videos already served by the service  $\mathcal{V}_{\text{known}}$ . More precisely, we compute for each tag  $t$  an average per-video and per-country “contribution” of this tag to the views of the known videos in which  $t$  appears: For this estimation, we use the training set to compute  $\mathbf{views}^{\mathcal{V}_{\text{known}}}(t)[c]$ , the aggregated number of views in country  $c$  of the videos of  $\mathcal{V}_{\text{known}}$  containing  $t$  as tag (Equation (3.5) from Section 3.1.2). From  $\mathbf{views}^{\mathcal{V}_{\text{known}}}(t)[c]$ , we then compute the average number of views in country  $c$  of the videos containing  $t$ :

$$\begin{aligned} \mathbf{views\_p\_vid}^{\mathcal{V}_{\text{known}}}(t)[c] &= \frac{\mathbf{views}^{\mathcal{V}_{\text{known}}}(t)[c]}{|\{v \in \mathcal{V}_{\text{known}} : t \in \text{tags}(v)\}|} \\ &= \mathbb{E}_{\substack{v \in \mathcal{V}_{\text{known}}: \\ t \in \text{tags}(v)}} \left( \mathbf{views}^{\mathcal{V}_{\text{known}}}(v)[c] \right) \end{aligned}$$

We then estimate  $\widehat{\mathbf{views}}(v)$  for  $v \in \mathcal{V}_{\text{test}}$  as:

$$\widehat{\mathbf{views}}(v)[c] = \mathbb{E}_{t \in \text{tags}(v)} \left( \mathbf{views\_p\_vid}^{\mathcal{V}_{\text{known}}}(t)[c] \right) \quad (4.3)$$

The placement works then as follows: we iterate over the videos of  $\mathcal{V}_{\text{new}}$ , and place  $R$  copies of each video  $v$  in the first  $R$  countries in which  $v$  is predicted to get most of its views, among the countries with some remaining storage.

### 4.2.3 Experiment, metrics, baseline

As in Section 4.1, we split our dataset in two, using the same reference ( $\mathcal{V}_{\text{train}}$ ), and testing sets ( $\mathcal{V}_{\text{test}}$ ).  $\mathcal{V}_{\text{train}}$  plays the role of known videos  $\mathcal{V}_{\text{known}}$ . Because



$\mathcal{V}_{\text{test}}$  remains particularly large (295448 videos, and 86,624,310,171 views), we sample it down while conserving the distribution of views across countries and tags. We first generate a trace  $\mathcal{T}$  of 10 millions requests for the videos of  $\mathcal{V}_{\text{test}}$  that respects the distribution of views between videos and countries. In other words, the probability to generate a request for video  $v$  in a country  $c$  in  $\mathcal{T}$  is proportional to the number of views of  $v$  in  $c$ :

$$P(\text{generate request}(v, c)) = \frac{\mathbf{views}(v)[c]}{\sum_{v' \in \mathcal{V}_{\text{test}}} \text{tot\_views}(v')}$$

We then choose  $\mathcal{V}_{\text{new}}$  as the set of unique videos present in the trace  $\mathcal{T}$ .

As baseline, we use a *random placement* policy, which randomly allocates each of the  $R$  replicas of a video in  $\mathcal{V}_{\text{new}}$  to any country with some remaining storage capacity.

We evaluate the quality of a placement by replaying the trace  $\mathcal{T}$ , and counting how often a request can be served from the country it originates from (a *hit*). In the case where we cannot serve the request (a *miss*), we store the video in the local country cache for future use under the LRU cache policy. We use the hit ratio ( $\#hits/(\#hits + \#misses)$ ) as our quality metric.

#### 4.2.4 Results

We start by comparing the average hit ratios obtained by our placement approach and by the baseline across all countries for different values of  $R$ . Results are shown in Figures 4.4–4.6. Figure 4.4 plots the average hit ratio obtained by each approach for different replication factors ( $R \in [1, 5]$ ). It shows that a tag-based placement clearly outperforms the baseline with an improvement that oscillates between 5.6% ( $R = 1$ ) and 6.8% ( $R = 5$ ). This advantage remains roughly constant as  $R$  increases, although for very large values of  $R$ , the difference between them will decrease. The two approaches will achieve the same 100% hit ratio when  $R$  will be so large that all videos could be stored in every country, obviously.

Figure 4.5 charts the cache performance of the 6 countries receiving the most views (US, India, Japan, Russia, Brazil and Great Britain), for three values of  $R$  (1, 3 and 5). The left bar above each country corresponds to the performance of the tag-based placement, and the right bar to that of the random placement. Each bar shows the absolute number of misses (top black line), of hits served by the LRU cache (middle red hatched section), and of hits served by the primary

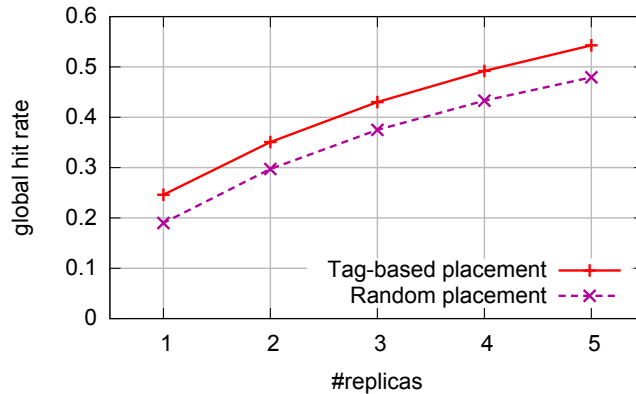


Figure 4.4: A tag-based placement strategy consistently improves the system’s global hit rate by about 6%, independently of the number of copies per video.

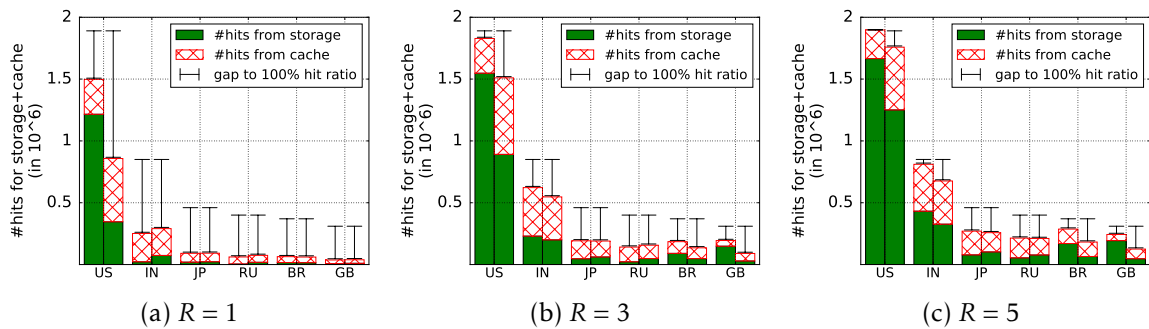
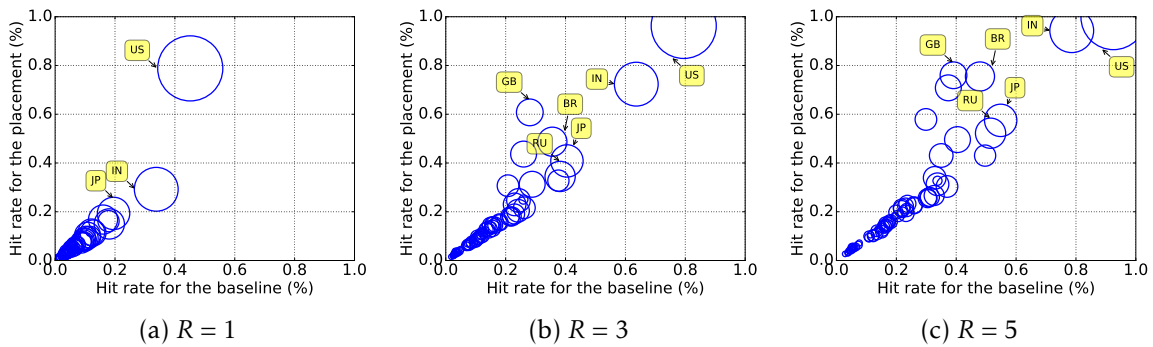


Figure 4.5: Hits and misses for the top 6 countries for  $R \in \{1, 3, 5\}$ , for the tag-based (left bars) and random placement (right bars). The green and red portions denotes respectively the contributions of the permanent storage  $S_c$  and of the cache  $C_c$ .

storage (bottom green solid section). The results show that tag-based prediction provides the most advantage for countries that view the most videos. For  $R = 1$ , the US obtain a hit ratio of 79% with our model and only 45% with the baseline. The composition of this hit ratio also changes: our approach achieves 64% of hit ratio through the primary storage and only 15% through the LRU cache; the baseline achieves only 18% through the primary storage and as much as 27% through the cache.

Increasing the number of replicas,  $R$ , yields an improvement for both approaches in every country. However, our tag-based placement ends up providing better results in all countries except Russia. This likely results from the fact that we had to ignore a large number of Cyrillic tags from our dataset.

Figure 4.6 provides a different perspective on the results over the entire set of countries. Each circle in the figure represents a country and its surface is pro-



**Figure 4.6: Hit ratios obtained through random placement (x-axis) vs. tag-based placement (y-axis) for all country (individual bubbles) for  $R \in \{1, 3, 5\}$ . The bubbles' area shows the number of views of individual countries.**

portional to the number of requests from that country. The  $x$  axis represents the accuracy of the baseline approach and the  $y$  axis represents that of the tag-based approach. The figure shows that the countries with the biggest share of views benefit the most from our solution. With  $R = 1$  the improvement remains limited to the US, but as  $R$  increases more and more countries see an improvement in their results. For  $R = 5$ , we also observe that the US reaches almost 100% with both approaches. This clearly limits the improvement that can be provided by any protocol.

Both Figures 4.5 and 4.6 show that tag-based placement works best for countries with many views, and that the number of benefiting countries increases with  $R$  (from one—the USA—with  $R = 1$ , to more than ten and twenty respectively for  $R = 3$  and  $R = 5$ ). This phenomenon directly results from our greedy placement algorithm: countries with many views are predicted more often as a top country and thus attract more “good” primary copies. With a single copy per video ( $R = 1$ ), runner-up countries (such as India, or Japan) are thus prevented from storing videos that would be good matches for their viewership, because these have been preferentially attracted to the US. When  $R$  increases this phenomenon moves down the list of countries. The overall effect remains an average increase in hit ratio (Figure 4.4).

### 4.3 Summary

The approaches and results presented in this chapter show that the tags attached to videos can be exploited to predict where a new video might be most

viewed. The approaches we have proposed are simple and cheap (relying on sums and averages), while delivering a reasonable accuracy. Our second experiment also demonstrates that such predictions can help place new videos in an intelligent manner, in particular in distributed delivery system with a limited amount of resources, and a globally distributed infrastructure.

One limitation however of the placement mechanism we have presented is that the computation of view predictions assumes that the view distribution of previous videos and previous tags is globally known to all participants in the storage system. This assumption is difficult to implement in an efficient and scalable manner in a fully decentralized systems. **In the next chapter, we therefore turn to the problem of estimating the viewing potential of a new video using a decentralized prediction protocol in order to remove this limitation.**



# Decentralized Estimation of Geographic Video Views

## Contents

5.1	Motivation and general system architecture . . . . .	65
5.1.1	System model . . . . .	65
5.1.2	Background: Self-organizing overlays . . . . .	67
5.1.3	A decentralized placement protocol . . . . .	68
5.2	Mignon: Fast Sum Estimation . . . . .	71
5.2.1	Trapezoidal rule. . . . .	72
5.2.2	Gregory-Newton Interpolation. . . . .	73
5.2.3	Polynomial Least Squares Fit. . . . .	73
5.3	Mignon Evaluation . . . . .	74
5.3.1	Accuracy Comparison . . . . .	76
5.3.2	Mignon Sensitivity Analysis . . . . .	78
5.4	Discussion . . . . .	86
5.4.1	Influence of Sample & Collide . . . . .	87
5.4.2	Convergence speed . . . . .	87

5.5 Summary . . . . . 88

In this chapter we put into practice part of the results of the previous chapter on tag-based placement in the context of a decentralized architecture. Our results suggest that the placement approach of Chapter 4 holds the potential to improve the behavior of decentralized storage systems for UGC videos. This placement approach requires however to compute the likely geographic view distribution of each new video. In a truly decentralized system, such as the ones we advocate in this thesis, this computation itself should be decentralized. The question of how such a decentralized view prediction should occur is the one we turn to in this chapter.

## 5.1 Motivation and general system architecture

### 5.1.1 System model

We consider a global decentralized P2P UGC service, in which each user contributes her resources to the system, whether it be a set-top box, a connected device or a personal computer. As we focus on content placement—explicitly here on videos—and view prediction, we assume our service can store and retrieve videos from users' machines in a decentralized manner [60, 62, 54], but we do not detail these mechanisms any further in the rest of the chapter. As is now common in many on-line services, we also assume that each user who desires to upload content to the system is free to attach a set of tags of her choice, limited neither by the size of the set nor by the nature of tags itself. Each node keeps track locally of the previously consumed content. More precisely, the individual devices of users (label 1) store the list of videos they have consumed (their *video profile*, label 2). Each video is associated with a set of descriptive tags provided by its uploading user [35, 33] (label 3). The tags of the videos viewed by a user form her *tag profiles* (label 4). We rely on a tag-based *affinity function*,  $f$ , that measures a user's affinity with new videos (5) based on her previous consumed videos. A number of such rating functions exist, from cosine similarity [16] to average views per tags [24]. The only assumptions we make about  $f$  is it uses the tags profiles of individual users and of the new video, and its result is correlated with the probability that this user will watch the video (6).

When uploading a new video, copies of this video should ideally be placed in storage locations close to where it might be most consumed. This is because the viewing patterns of many videos in UGC services present clear geographic



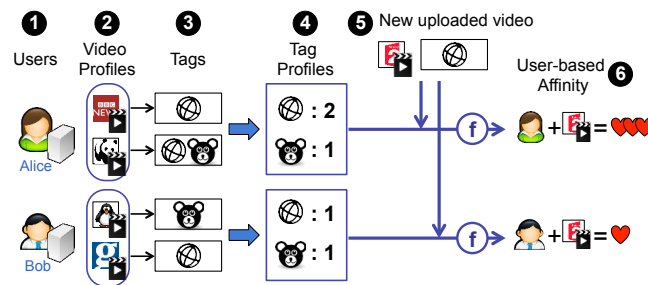


Figure 5.1: Using tags to predict users' affinity with a new video

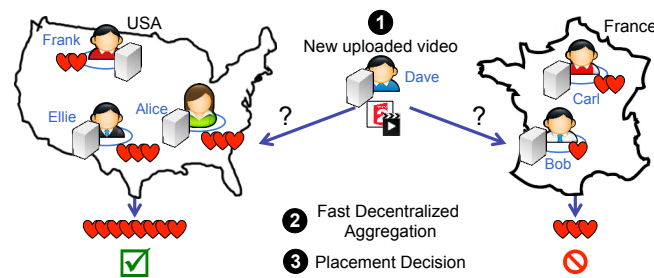


Figure 5.2: Placing new videos based on aggregated affinity

trends [18], which are strongly correlated with a video's tags as analyzed in the previous chapter. In Fig. 5.2 for instance, Dave must decide whether to store his new video in the USA or in France. The decision process has to be determined by the video's likely future popularity in the different countries, which can be estimated by the sum of all user affinities in each country. Obtaining this aggregated sum efficiently is unfortunately challenging in a large P2P system. Dave could trigger a P2P aggregation in the USA and France [50], but such an approach would require computing the similarity between the new video and every user in each country, a slow and costly operation.

In this chapter, we therefore investigate how such a sum can be efficiently, rapidly, and accurately estimated in a fully decentralized system while involving only a small subset of the users in a given country. Our proposal exploits the capabilities of similarity-driven self-organizing overlays to estimate the aggregated affinity of a new video within a community of users (here located in the same country) while avoiding a full-fledged decentralized aggregation procedure.

Instead of launching an expensive aggregation every time a new video is uploaded, we propose a cheaper mechanism to estimate the aggregated affinity of a video. Our approach exploits a similarity-driven overlay [16] that interconnects all the users in a country. In the following we first briefly describe similarity-driven overlays, and then present the details of our approach.

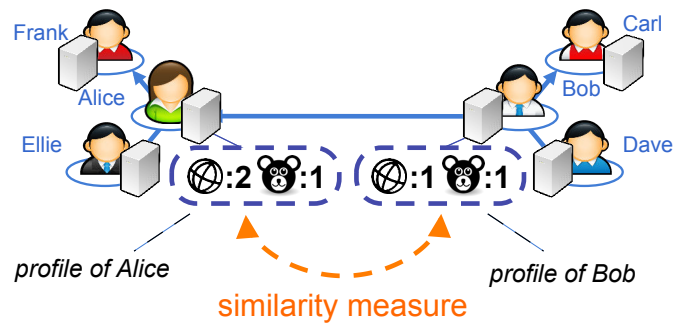


Figure 5.3: A self-organizing overlay

### 5.1.2 Background: Self-organizing overlays

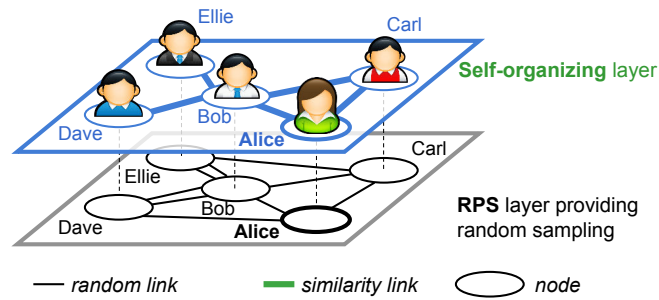
Peer-to-peer overlay networks extend the capabilities of an underlying network (e.g. based on TCP-IP) with additional indexing and routing capabilities to provide richer services, typically in user space. They do so by organizing a large number of machines (known as *peers* or *nodes*) in a flat topology in which each peer is connected to a small number of other peers (known as the *view* or *neighborhood* of the peer).

The scalability and robustness of peer-to-peer overlays have made them well adapted to large scale distributed systems such as decentralized social networks.

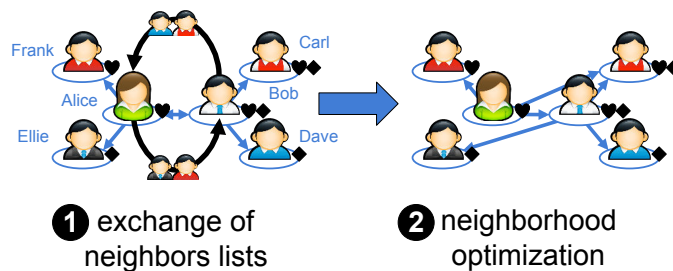
Similarity-driven overlay networks organize peers according to their similarity between the interests of their associated users [42], with a wide range of data-oriented applications [16, 14, 17, 31, 29] like decentralized query expansion [16], peer-to-peer search [14], news recommendation [17], and CDNs [31, 29]. In this thesis, we consider gossip-based similarity driven overlays, whose behavior is illustrated in Figures 5.3-5.5.

The machine of each user (e.g. a user's set-top box) holds the user's profile: in our case the list of viewed videos and their attached tags (Fig. 5.3). Alice has viewed two videos associated with the tag 'World', and one associated with 'Animals'. Bob has viewed one video associated with 'World', and one associated with 'Animals'. Starting from random neighbourhoods (which depend on how each node joined the network) the goal of the overlay is to eventually connects each peer to its  $k$  most similar other peers in the network, according to some similarity metric (e.g. Jaccard's coefficient, or Cosine Similarity). We use cosine similarity in the following.

This neighbourhood construction uses two epidemic protocols executing on each peer (Figures 5.4 and 5.5). With the first mechanism, a peer (e.g. Alice) reg-



**Figure 5.4: Overlay Architecture**



**Figure 5.5: Peer-to-peer neighborhood optimization**

ularly polls an underlying and constantly evolving Random Peer Sampling (RPS) overlay [43] to obtain a set of random peers from the rest of the system. In Fig. 5.4 for instance, Alice might discover Dave through the RPS layer. If Dave turns out to be a better neighbour for Alice than Bob (upper self-organizing layer), Alice will replace Bob by Dave in her neighbourhood. This stochastic process ensures that, if all user profiles remain equal, the system eventually converges to an optimal state. In large networks the convergence might however be very slow. To speed up convergence, peers use a second ‘*neighbor-of-neighbor*’ mechanism (Fig. 5.5). The intuition is that if Alice is similar to Bob, and Bob to Carl, then Carl might be similar to Alice. Peers therefore periodically exchange their current neighbours lists (Step 1 in Fig. 5.5), and use the new peers they discover to optimize their neighbourhoods (Step 2). This mechanism greatly accelerates convergence (usually in  $\log(N)$  rounds [42]), where  $N$  is the size of the network, but might get stuck in a local minimum, and is therefore complementary to the stochastic mechanism of Fig. 5.4.

### 5.1.3 A decentralized placement protocol

In order to implement our sum estimation mechanism in a fully distributed storage system, each node runs the exact same architecture than any node present

---

**Algorithm 1:** Algorithm of the daemon running on each node

---

```

1 if TimeOutLocalKNN then
2   | LocalKNNTimer  $\leftarrow$  default
3   | viewlocal  $\leftarrow$  MaintainLocalListofPeers()
4   | ExecuteLocalKNN(ownprofile)
5 end
6 if TimeOutGlobalView then
7   | GlobalViewTimer  $\leftarrow$  default
8   | viewglobal  $\leftarrow$  MaintainGlobalListofPeers()
9 end

```

---

in the system. This architecture is structured on two distinct levels, each being responsible of one of the two main roles described previously in Section 5.1.2. Each node encompasses these two different levels in his permanently running daemon, described in Algorithm 1. In an idle state, each node periodically executes a maintenance routine on these two elements.

The first main element (lines 1-5) is executed everytime *LocalKNNTimer* comes to its end and signals it via *TimeOutLocalKNN*. Its role is to maintain a similarity-based overlay network of peers inside a country (*local level*), allowing each peer to keep in its local list of neighbours (*view<sub>local</sub>*) the  $k$  peers having the higher similarity measure with its *ownprofile*. The similarity between two peers is computed by having them exchange their personal record composed of the different tags encountered in the contents consumed by the peer so far, and the respective number of occurrences of each of these tags, englobed in a personal profile, denoted *ownprofile*. A Random Peer Sampling (RPS) service is used on this *local level* in order to avoid local minimum as described in Fig. 5.4.

The second main element in Algorithm 1 is triggered by *TimeOutGlobalView* and act on an inter country overlay or *level* (*global level*). This element is responsible to maintain a global view of the system denoted *view<sub>global</sub>* allowing each node to be in contact with a least 1 peer in each different country. For this end, it rely on a RPS service, operating over a geographic overlay. We assume that geographic overlay, not detailed in this thesis, allows each node to contact at least one peer in each various geographic zone *local level*, with a given static zone identification, such as via a prefix in the node identification allowing to broadcast a message to the peers in a particular country.

Apart from the daemon running continuously inside each connected node,

---

**Algorithm 2:** Algorithm for the Upload and Placement parts
 

---

```

1 on receive upload(file) request do
2   |  $P_v = \text{CreateVirtualPeer}()$ 
3   |  $world \leftarrow \emptyset$ 
4   |  $world = \text{RequestCountriesKNN}(P_v, view_{global})$ 
5   |  $bp \leftarrow \text{ExtractBestPlaces}(world, R)$ 
6   | ask  $store(file.info)$  to  $bp$ 

```

---



---

**Function** RequestCountriesKNN
 

---

```

Input:  $P_v, view_{global}$ 
Output: Partial Similarity graph for each country
1 from  $view_{global}$  select  $Nodes$ 
2 for  $node$  in  $Nodes$  do
3   |  $\text{SendKNNOrder}(node, P_v)$ 
4 end
5 while not all Nodes responded and !TimeOutAsnwers do
6   |  $world \leftarrow \text{ReceiveAnswerGraph}()$ 
7 end
8 return  $world$ 

```

---

and maintaining for each node the two overlays, each peer have the ability to upload a file and place it to the geographical area where it will be the most consumed is described in Algorithm 2.

When a peer uploads a video  $v$ , she first attaches a set of self-defined tags to it, and use these tags to build a virtual peer  $P_v$ , whose profile contains the user-defined tags and attributing of value of 1 for their number of occurrences. As detailed in Function RequestCountriesKNN the uploader sends  $P_v$  to at least one node in each existing geographic area by relying on the peers comprised in  $view_{global}$ . It receives in return from each contacted foreign node an estimation of the corresponding country interest towards the profile of  $P_v$ . Based on these estimations,  $R$  best geographic areas are extracted, with  $R$  being the replication level of the system, and a storage request is send to each previously contacted node in these elected best places. In case an elected node for various reasons, could not store the content, he will rely on his own  $view_{local}$  to relay the storage request to the next interested peer, as described in Algorithm 3.

To compute the interest of each country towards  $P_v$ , we simply estimate the sum of the similarities between  $P_v$  and every other user in the country. To compute this sum exhaustively, either at peer  $P_v$  or using a standard aggregation pro-

protocol, we would either have to collect the profiles of all other nodes at  $P_v$ , or disseminate the profile of  $P_v$  to every other node. In both cases, the delay and the resulting network cost would be prohibitive for very large networks.

Instead we propose that the uploading user simply impersonates the virtual peer by having it join the similarity-based overlay of each country, by sending a similarity-measure request detailed in Algorithm 4 to one peer in each country, relying on peers present in  $view_{global}$ . If convergence in a similarity-based overlay is generally fast (generally logarithmic in the size of the network [42]), it is even faster for  $P_v$  to converge to obtain its  $k$ -nearest-neighbours in an already converged network. Once this happens, the uploading user exploits the content of the KNN and RPS neighbourhoods of  $P_v$  to estimate the video's aggregated affinity without any further network exchanges. This last element uses a new decentralized sum estimation protocol we have termed Mignon, which allows a peer to estimate the aggregated interest of a group of peer is central in our architecture, and is introduced in more detail in the following section.

## 5.2 Mignon: Fast Sum Estimation

In this section, we present the key contribution of this chapter: Mignon, a decentralized protocol for efficiently estimating the sum of a set of values. The key feature of Mignon consists in considering the affinity values of users found in the KNN and RPS views of  $P_v$  as samples taken from a monotonically decreasing function. Mignon uses these values to interpolate the function's shape, from which we derive an aggregated affinity by integration. The values obtained from the KNN neighbours constitute the first  $k$  consecutive samples, while those in the RPS represent randomly chosen samples distributed along the rest of the x-axis. To associate each of them with an x-coordinate (which the RPS does not indi-

---

### Algorithm 3: Algorithm handling the Request for Storage

---

```

1 on receive store(file.info, file.size) request do
2   if free space > file.size then
3     | StoreFile(file.info, requester)
4   end
5   else
6     | ask store(file.info, file.size) to bestneighbor
7   end

```

---

---

**Algorithm 4:** Algorithm for the estimation of a local interest
 

---

```

1 on receive CountryKNN request do
2   |  $prof s \leftarrow ExecuteLocalKNN(P_v)$ 
3   |  $CreateTrendGraph(prof s)$ 
4   |  $SendGraph(prof s, requester)$ 

```

---

cate), we rely on a network-size estimation protocol [46] that provides us with the length of the x-axis, and assume that the RPS samples are equally spaced along this axis.

We observe that the need to rely on a size-estimation protocol does not offset the benefits of Mignon in terms of delay and network cost. First, the size-estimation protocol does not need to be run for every video upload. Rather, in a setup consisting of set-top boxes that are almost always on, the protocol can run every few days. Second, protocols like Sample & Collide [46] can estimate the size of the network within a reasonable error margin at a minimal cost. We evaluate the impact of protocols like Sample & Collide in Section 5.4.1.

In the following we focus on three different techniques we use in Mignon that can estimate the unknown affinity values based on the KNN and RPS samples. We then evaluate their accuracy levels in Section 5.3.

### 5.2.1 Trapezoidal rule.

The first technique we consider is the trapezoidal rule, a well-known method for approximating the integral of a function. The rule replaces the function to be integrated with a sequence of linear segments and computes the integral as the sum of the areas of the corresponding trapezoids.

$$\int_a^b f(x) dx \approx \frac{1}{2} \sum_{k=1}^{N-1} (x_{k+1} - x_k)(f(x_{k+1}) + f(x_k)) \quad (5.1)$$

In our case, we do not have a function defined over a continuous interval of real number, but a discrete set of score values. As a result, in addition to the estimation error associated with the trapezoidal rule, our estimate will also include an error that results from the imprecise linearised placement of the samples taken from the RPS view. We evaluate this impact in Section 5.3.

### 5.2.2 Gregory-Newton Interpolation.

As a second estimation mechanism, we consider a polynomial interpolation. Specifically, we compute the polynomial of degree  $n - 1$  that goes through all of the  $n$  samples in the KNN and RPS. We then use this polynomial to compute the values associated with the users that are not among the samples. The Gregory-Newton Forward difference approach requires to calculate a difference table. This table is filled by the differential operator  $\Delta f_i$ . For the first order, this operator is :

$$\Delta f_i = f_{i+1} - f_i$$

and towards the  $k$  order:

$$\Delta^k f_i = \Delta^{k-1} f_{i+1} - \Delta^{k-1} f_i$$

The order is given by the number of points  $n$  used for the interpolation with  $k$  equals  $n - 1$ . The polynomial equation look like:

$$\begin{aligned} P_k(x) = f_0 + \Delta f_0 \frac{x - x_0}{h} + \frac{\Delta^2 f_0}{2!} \frac{x - x_0}{h} \left( \frac{x - x_0}{h} - 1 \right) + \dots \\ \dots + \frac{\Delta^k f_0}{k!} \frac{x - x_0}{h} \left( \frac{x - x_0}{h} - 1 \right) \dots \left( \frac{x - x_0}{h} - k + 1 \right) \end{aligned} \quad (5.2)$$

with  $h$  being the step between two samples, and must remain constant, such as  $x_1 = x_0 + h$  and  $x_n = x_0 + nh$ . To solve this condition, we will first apply the trapezoidal rule to integrate the values of the k-firsts interests, these values given by the KNN having a constant spacing of 1 between them, then apply this interpolation to the linearised interval of the scores given by the RPS service, the linearisation allowing us to maintain a constant gap between two measures. We will then add the two area interpolated.

### 5.2.3 Polynomial Least Squares Fit.

Our last technique in Mignon is to consider a least squares regression on a degree-two polynomial. In short, we compute the degree-two polynomial coefficients that minimize the square error with respect to the samples in the KNN and RPS, and sum the values that the polynomial associates with each of the users in the system.

This method try to fit the given matrix of coordinates inside a polynomial. The form is:



$$\begin{aligned}
p(x) &= ax^2 + bx + c \\
s.t.(a, b, c) &= \arg \min_{(a,b,c)} \sum_{j=0}^k |p(x_j) - y_j|^2
\end{aligned} \tag{5.3}$$

The choice of using a degree-two polynomial was motivated by the general shapes of the graphs of interests, as we considered them sorted, we obtain a graph representing a decreasing function.

### 5.3 Mignon Evaluation

We evaluate Mignon on two distinct datasets. The first consists of the YouTube dataset we introduced and analysed in Chapter 4. To evaluate Mignon, we “reinterpreted” this dataset by considering each country as if it was a single user. Our modified dataset therefore consists of 257 users in a single country with the interest rating of a video for each country being the number of views in this country.

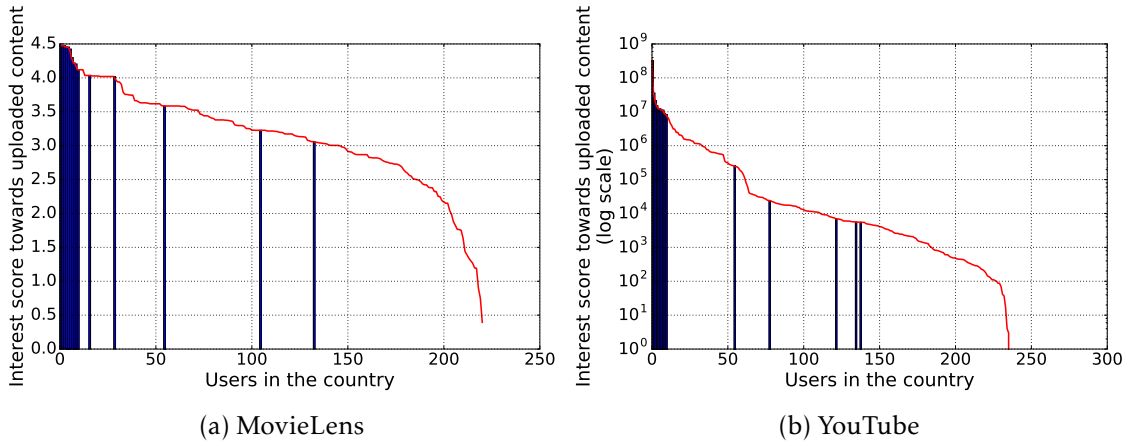
Our second dataset, MovieLens, consists of a trace from a personal movie recommendation system<sup>1</sup>. MovieLens started as a project from GroupLens<sup>2</sup>, a well known research lab in the Department of Computer Science and Engineering at the University of Minnesota, from whom the datasets are widely used in experimental research on recommender systems. It contains a set of 22,000,000 ratings and 580,000 tag occurrences applied to 33,000 movies by 240,000 users. The second part of the dataset is denoted as a tag genome, and contains 11,000,000 computed tag-movie relevance scores from a pool of 1,100 tags applied to 10,000 movies. Each movie is associated with a vector of ratings (1 to 5 integers) by a subset of the users, and a set of  $n$  pairs, each consisting of a tag and a real-valued relevance score. The rating,  $R_u(m)$ , expresses the rating of a movie  $m$  by a user  $u$ , while the relevance score,  $r_m(t)$ , expresses the importance of a tag,  $t$ , for a given movie,  $m$ . Based on this information, we compute the interest score  $u_t$  of a user  $u$  for a tag  $t$  by averaging the product of rating and relevance over all the  $N_u$  movies seen by user  $u$  as follows.

$$u_t = \frac{1}{N_u} \sum_{m=1}^{N_u} (r_m(t) * R_u(m)) \tag{5.4}$$

---

<sup>1</sup>[www.movielens.org](http://www.movielens.org)

<sup>2</sup><http://grouplens.org/datasets/movielens/>



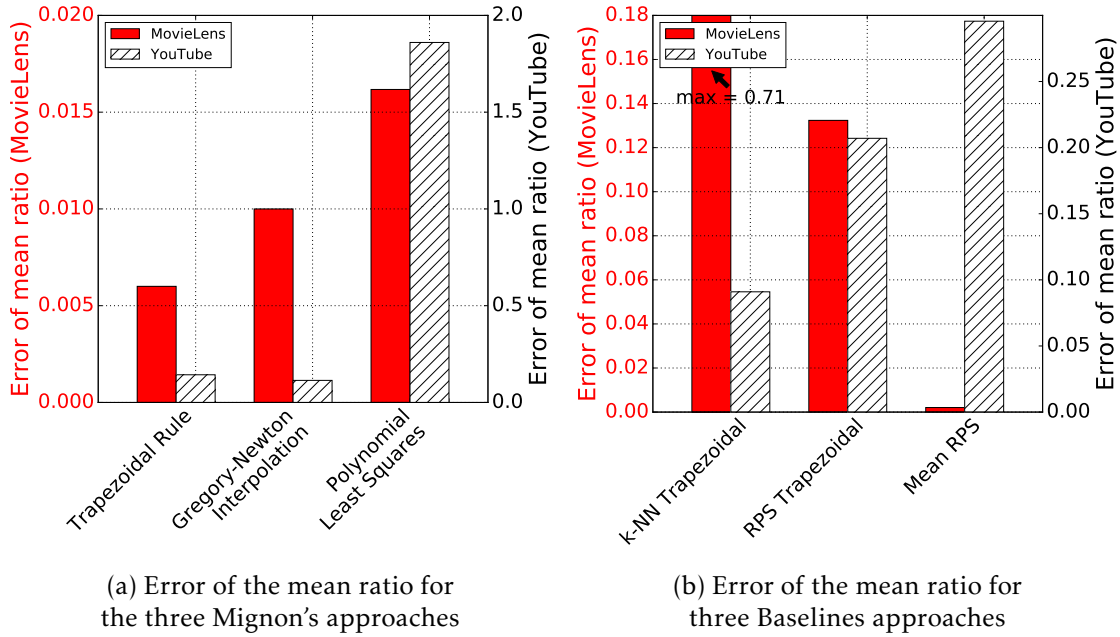
**Figure 5.6: Interest curve for MovieLens(a) and YouTube(b) datasets. Black vertical lines represent KNN and RPS samples.**

Since we want to evaluate Mignon’s ability to estimate the aggregation of a score value, we consider a synthetic set of new “videos”, whose profile only comprises a single tag taken from the dataset.

Since we want to evaluate Mignon’s ability to estimate the aggregation of a score value, we use two approaches for each dataset. For MovieLens, we select all the tags available and consider that each of them plays the role of an hypothetical new video that would only have this particular tag attached to it. For each of these tags, we build an appreciation profile for each of them, from the aggregated users interests mentioned before, obtained via Equation 5.4, and then consider a set of “videos” embedding each one a tag from the above set. We use the same tag-profile approach for YouTube dataset only this time we only consider a randomly picked subset of 1,000,000 tags embedded in at least 1,000 different videos.

For each such video  $v$ , we first select the set of users in its KNN and RPS views, and then compute its affinity with these users. We use this sample of affinity values to produce an estimate (noted  $\hat{a}_v$ ) of the video’s aggregated affinity with all the users in the system (which we note  $a_v$ ). To assess the performance of different estimation techniques, we define an estimation ratio:  $ER_v = \frac{\hat{a}_v}{a_v}$ . We evaluate  $ER_v$  in a variety of configurations on each of our datasets. Let  $n$  be the number of tags in a dataset (and hence of synthetic videos), we present the distribution of  $ER_v$ , its mean  $\overline{ER} = \frac{1}{n} \sum_{i=1}^n ER_{v_i}$ , as well as its standard deviation  $\sqrt{\overline{ER^2} - \overline{ER}^2}$ .

Figure 5.6 exemplifies the affinity score distribution of particular tags (interpreted as videos) in each of the two dataset. The curve depicts the affinity score of each user for the tag in decreasing order, while the vertical bars represent the



**Figure 5.7: Evaluation of the error for both datasets MovieLens and YouTube**

data available in the KNN and RPS views.

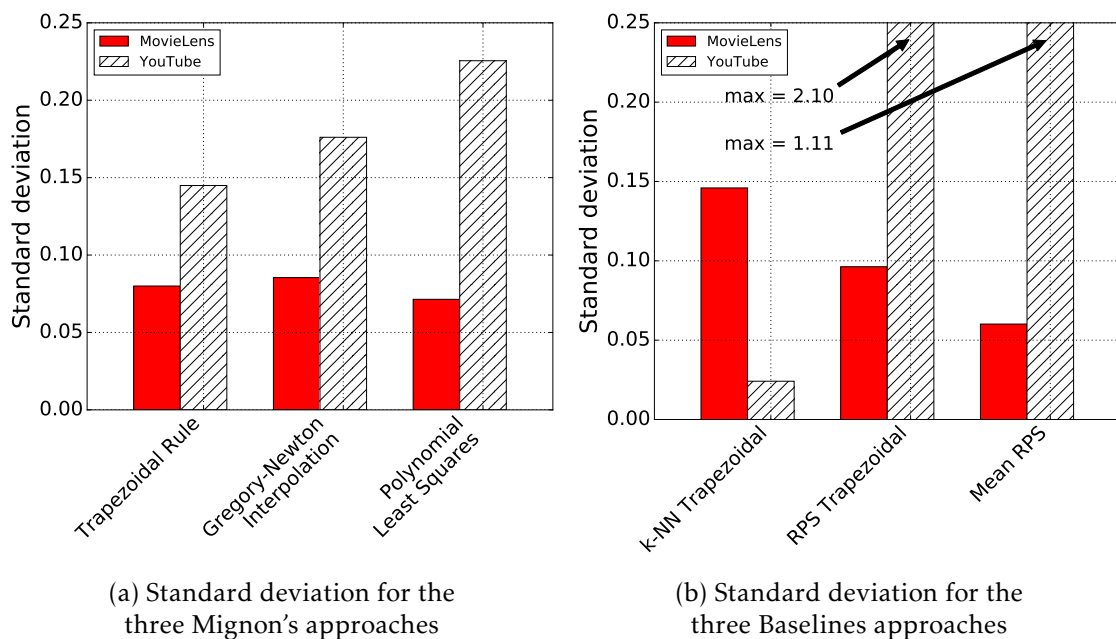
### 5.3.1 Accuracy Comparison

We start our evaluation by comparing the results obtained by Mignon with those obtained by three baseline approaches that exploit either the KNN or the RPS views but not both. For Mignon, we consider the three estimation techniques presented in Section 5.2 (the *Trapezoidal* and *Gregory-Newton* interpolations along with *Polynomial Least Squares* regression). For the baselines, we tested both these techniques as well as linear and quadratic regression and selected the three that obtained the best performance.

Specifically, **KNN-Trapezoid** applies the trapezoid rule on a KNN view without using the RPS, **RPS-Trapezoid** also applies the trapezoid rule but on an RPS view with no KNN, while **RPS-Mean** simply computes the average similarity of the nodes in the RPS view and multiplies it by the size of the network. We configured our techniques to use a KNN view size of 15 and an RPS size of 10, while all the baselines use a single view (RPS or KNN) of size 25. Figure 5.7 shows the results on both of our datasets, depicting the error on the mean estimation ratio, that is  $|\overline{\text{ER}} - 1|$ , and shows that combining the KNN and the RPS views allows Mignon to adapt to multiple data sets. Specifically, all three

techniques, the Trapezoidal rule, Gregory-Newton interpolation and Polynomial Least Squares obtain very good estimates on MovieLens dataset with an error on the mean ratio respectively of 0.006 (0.6%), 0.01 (1%) and 0.016 (1.6%) on MovieLens. For YouTube dataset, although the Trapezoidal rule and Gregory-Newton interpolation performs well with an error of mean ratio respectively of 0.143 (14.3%) and 0.114 (11.4%), the Polynomial Least Squares approach performs poorly, with an error of 1.86 (186%), due to the inability of this regression in degree-two to deal with the skewness of YouTube's views distribution, as shown in Figure 5.6b. The baselines, on the other hand, can achieve good performance on one of the datasets but not on both. KNN-Trapezoid achieves a very low error of 0.09 (9%) on YouTube, but a very high error of 0.7 (70%) on MovieLens. RPS-Mean achieves a very low error of 0.02 (2%) on MovieLens but a high error of 0.30 (30%) on YouTube, while RPS-Trapezoid achieves errors of 0.13 (13%) on MovieLens and of 0.21 (21%) on YouTube, worse than both of Mignon's approaches on both datasets.

Figure 5.8 completes the picture by showing the standard deviation of the estimation ratio for both Mignon and the three baselines. Again, the three Mignon's techniques obtains low standard deviations on both data sets, with the Polynomial Least Square regression performing best on MovieLens (7%) and worst on



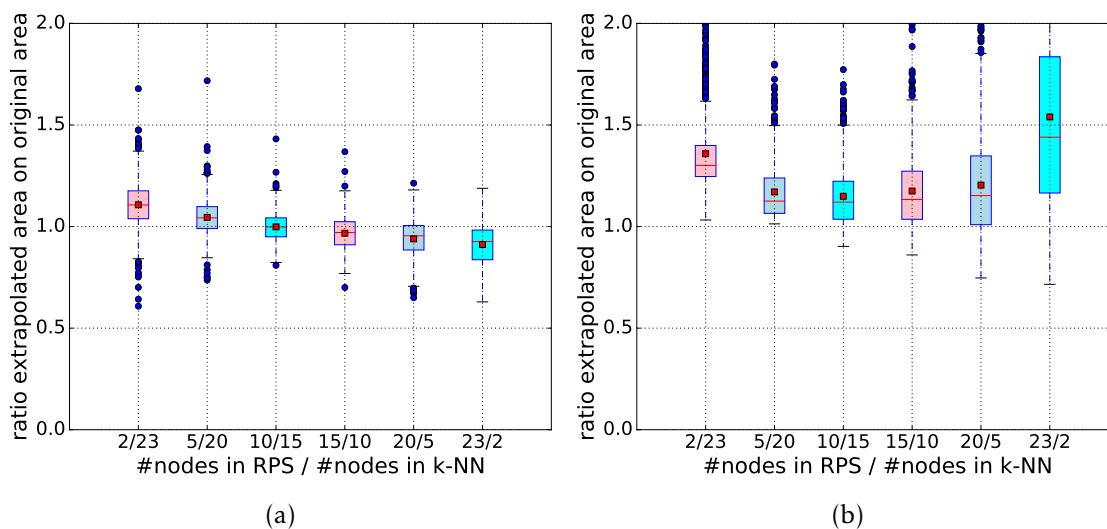
**Figure 5.8: Evaluation of the standard deviation for both datasets MovieLens and YouTube**

YouTube (21%), Trapezoidal rule performing best on YouTube’s dataset (14.5%) and Gregory-Newton interpolation scoring worst on MovieLens (8.2%). Regarding the baselines approaches, they generally scores worse than Mignon on both MovieLens and YouTube, with two notables exceptions. For YouTube, KNN-Trapezoid obtains a good standard deviation of 2.2%, far from Mignon’s best obtained by the Trapezoidal rule(14.5%), and for MovieLens RPS-Mean obtains a stadand deviation of 6% closely outperforming Mignon’s best with the Polynomial Least Squares reaching 7%.

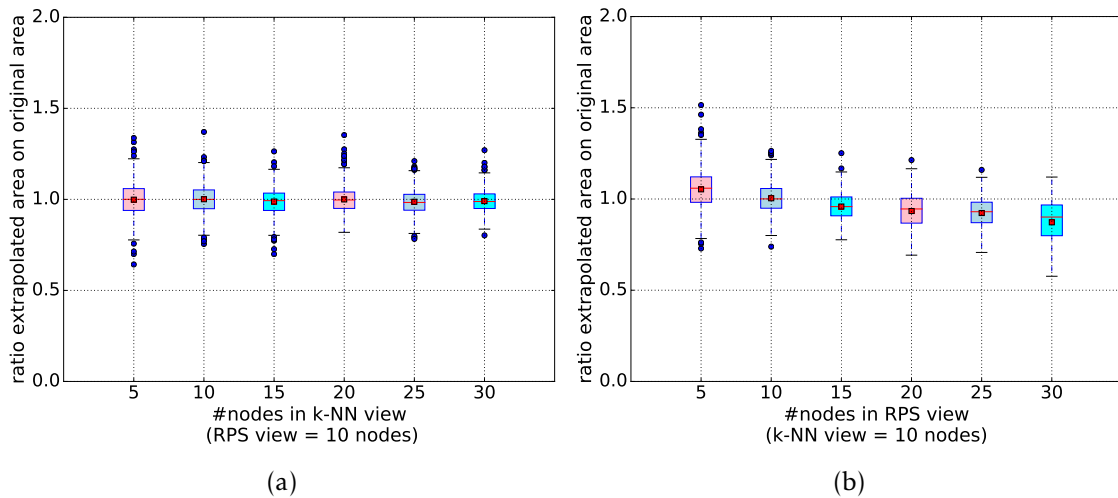
### 5.3.2 Mignon Sensitivity Analysis

Now that we have shown the effectiveness of Mignon’s estimation approach on multiple datasets, we analyse how the KNN and RPS views impact its performance. We present our results in the form of whisker plots. Each box in the plot covers the values between the lower and the upper quartiles; the point in the box represents the mean, while the line the median. The endpoints of the whiskers represent the lowest datum still within  $1.5 \times \text{InterQuartile Range (IQR)}$  of the lower quartile, and the highest datum still within  $1.5 \times \text{IQR}$  of the upper quartile, while the points outside the whiskers represent outliers.

**Trapezoidal rule.** Figure 5.9 shows how the effectiveness of the trapezoid rule varies when we vary the sizes of the KNN and RPS views. For fairness we main-



**Figure 5.9: Fast Decentralized Area Estimation using the trapezoid rule in the MovieLens dataset(a) and YouTube dataset(b).**

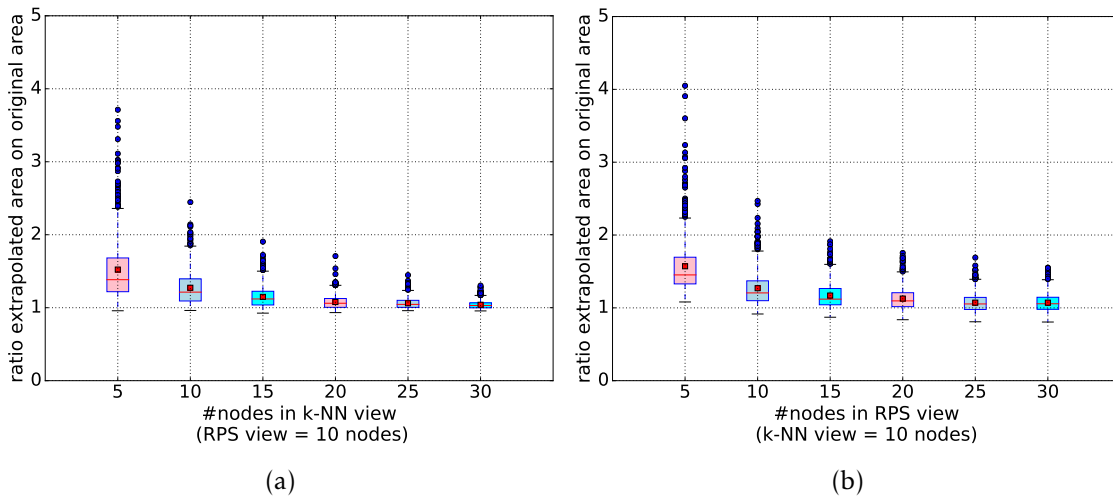


**Figure 5.10: Influence for the Trapezoid Rule of varying the size of the KNN(a) and RPS(b) for the MovieLens dataset.**

tain a total view size of 25 and vary the proportion of nodes in the two views from  $|\text{KNN}|=2$   $|\text{RPS}|=23$  to  $|\text{KNN}|=23$   $|\text{RPS}|=2$ . Figure 5.9a shows that larger KNN views slightly tend to overestimate the total affinity, while larger RPS views slightly tend to underestimate it, with the best performance being achieved with a KNN view of 15 and an RPS view of 10. We observe the good accuracy and the relative narrowness of the distribution of ratio extrapolated area on original area, already highlighted in Figure 5.7 and 5.8.

Figure 5.10 highlights that in the MovieLens dataset the estimation error depends primarily on the size of the RPS view. Figure 5.10a shows the results with increasing KNN sizes with an RPS size of 10. Results appear almost independent of the KNN size, even though larger sizes slightly reduce the variance. Even if not shown in the plot for the sake of clarity, we verified that KNN view sizes as low as 2 yield a good mean estimate, even though with a larger variance. Figure 5.10b examines instead the impact of the RPS view size with a fixed KNN of 10. Unlike in the previous case, the quality of the mean estimate heavily depends on the size of the RPS view. Once again, we remark that the best trade-off between accuracy and skewness of the distribution is achieved for Figure 5.10a and Figure 5.10b when the sum of the KNN and RPS sizes reaches 25, whether it be for a RPS size of 10 and a size of 15 for the KNN or the contrary.

Figure 5.9b complements the above results with the performance of the Trapezoid rule on the YouTube dataset. Again, we obtain the best performance with a KNN-to-RPS ratio of 3/2. With a KNN view of 15 and an RPS view of 10, the

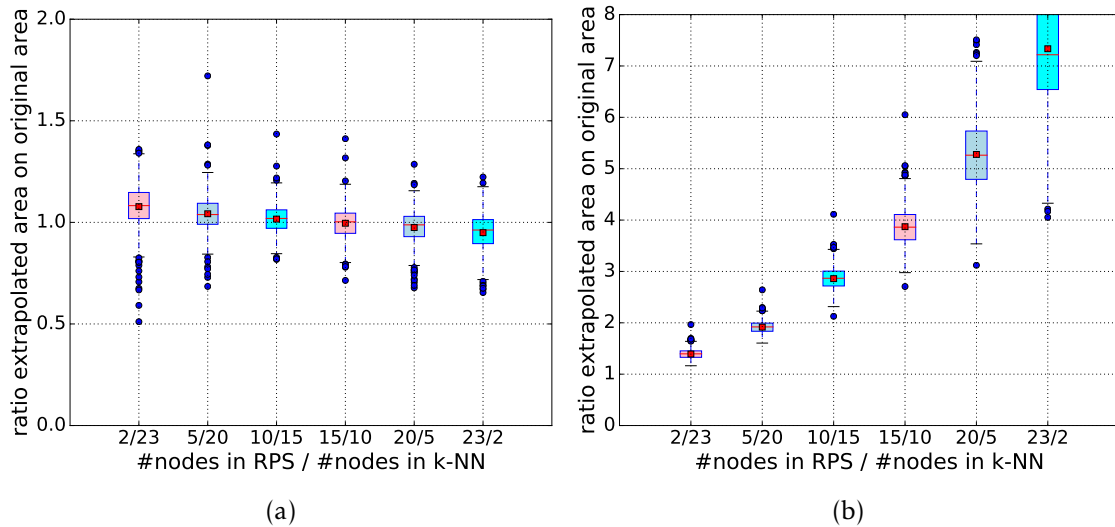


**Figure 5.11: Influence for the Trapezoid Rule of varying the size of the KNN(a) and RPS(b) for the YouTube dataset**

mean estimation ratio settles at 1.14. Moreover, slightly smaller or slightly larger KNN-to-RPS ratios impact this result only to a limited extent. In our tests, we observed that this results from the fact that when one view remains constant, performance consistently improves when increasing the size of the other.

These observations are detailed in Figure 5.11. The varying sizes of both RPS and KNN views have a direct impact on the accuracy of the estimation of aggregated interests. Although the mean values remains closely the same, we observe in Figure 5.11b that when keeping a constant KNN size of 10 and varying the RPS size, the accuracy of the estimation improves with the size of the RPS view, with the resulting estimations being more scattered than in Figure 5.11a, implying a higher variance, with a notable exception with low values of RPS size. The global aspect remain the same when keeping the RPS size at 10 and varying the KNN size. Figure 5.11a shows that varying the size of the KNN has a similar impact. A KNN view of 5 results in a significant estimation error of 1.5 times, while KNN views of more than 20 yield mean error ratios of less than 7%. The estimation error decreases with increasing view sizes but to a higher extent than in the case of the RPS view (mean error ratio of 11% with a view size of 20).

**Polynomial Least Squares regression.** Next, we evaluate the sensitivity of our second approach, the Polynomial Least Squares regression model. The results are displayed in 5.12, and as already observed in the previous paragraph in Figure 5.9, this method produce two distinct behaviours for the different datasets.

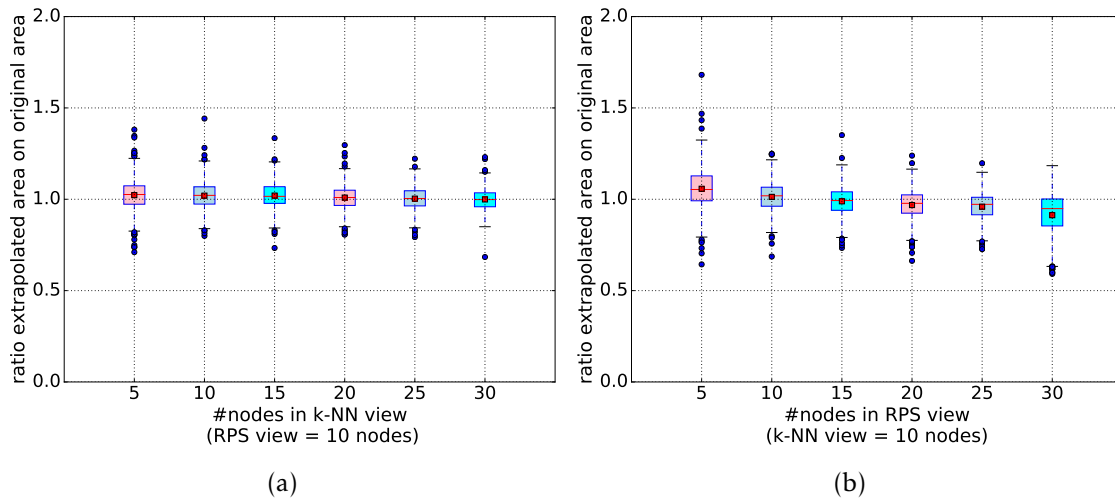


**Figure 5.12: Fast Decentralized Area Estimation using Polynomial Least Squares regression in the MovieLens dataset(a) and YouTube dataset(b).**

When applied to MovieLens as detailed in Figure 5.12a, the second approach behave exactly like the Trapezoidal rule, with an extrapolated aggregated interests area slightly superior to the original when the size of the KNN view is superior to the one of the RPS, and tending to the contrary when we increase the size of the RPS to the detriment to KNN. In that case however, the overall dispersion of the ratios have a smaller amplitude around the value of 1, indicating a more overall accurate estimation than the method of the Trapezoidal rule. The best results are obtained this time for a couple of values of 15 for the RPS and only 10 for the KNN, leading to a mean error ratio of only 0.4% and a median one of 0.26%. The application of this method for the YouTube dataset is illustrated in Figure 5.12b and scores very poorly compared to the previous method. As expected, the regression in this case is limited by it's low degree-depth of 2, and cannot cope with the steepness of a YouTube's distribution. The best results are achieved for having the largest size of KNN view, as when ordered decreasingly, the appreciation scores diminish drastically fast. The estimation error increase very quickly with the size of the RPS view is growing, to reach a mean error ratio of 734% for a KNN size of 2 and a RPS one of 23, compared to 53% for the same settings in the Trapezoidal rule. This settings being the worst case for both methods. The impossibility for this case of regression to fit the curve makes it non suitable for this type of distribution.

The detailed observation of the influence of the settings for this approach

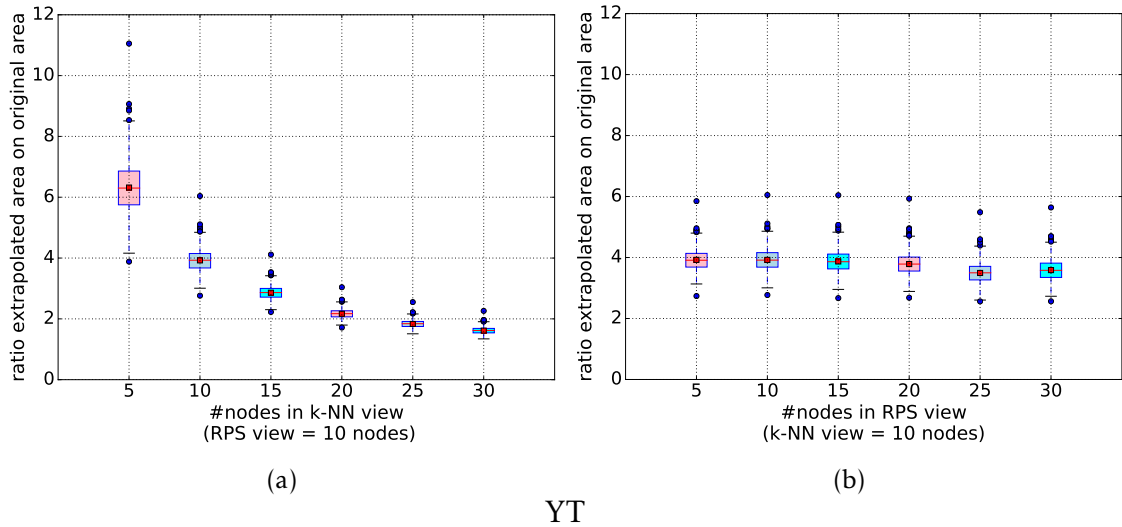




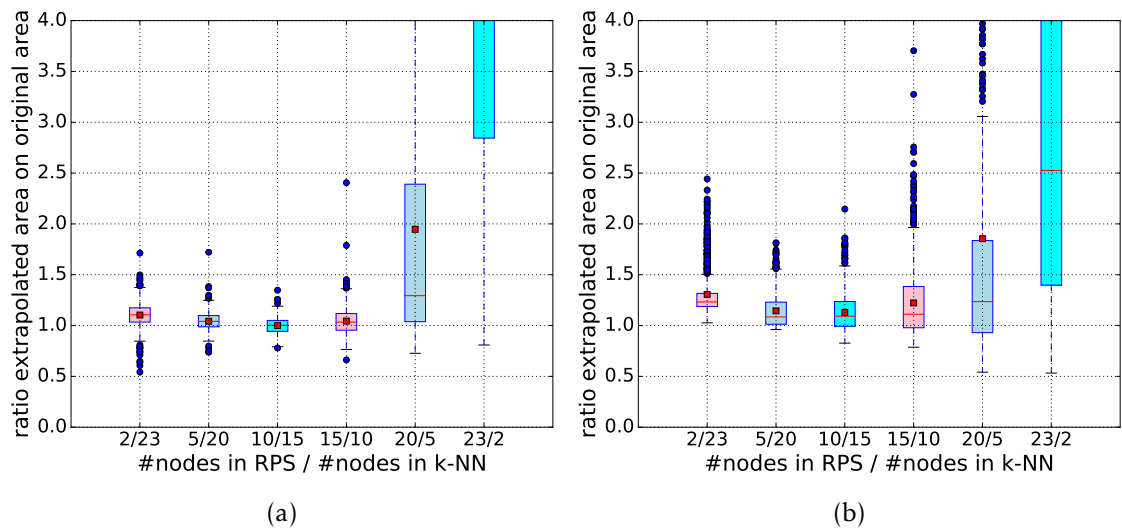
**Figure 5.13: Influence for the Polynomial Least Squares regression of varying the size of the KNN(a) and RPS(b) for the MovieLens dataset.**

when applied to a MovieLens dataset is detailed in Figure 5.13. Again, we witness in Figure 5.13a the almost non-existent influence of the KNN size when studied with a minimum value of 5, the difference between the lower and the upper quartile of the distribution of the ratios being close to constant. On the other hand, the overall decreasing attitude observed in Figure 5.12a is caused mostly by varying the RPS size, as described in Figure 5.13b.

When analysing the influence of the settings for YouTube's dataset, the results presented in Figure 5.14 reveal a much larger difference of impact between the KNN size and the one of the RPS. As it has been slightly hinted before in Figure 5.12b, we observe in Figure 5.14a the more neighbours we have in our KNN view, the more we are able to compensate the difficulty of the regression to fit such a tight curve, offered by a YouTube's distribution. We also observe the shrinking of the dispersion of the ratios along with the diminishing mean error when increasing the KNN size, tending towards the ratio value of 1. This observation coupled with the ones ensuing from Figure 5.14b highlight the inefficient role of the RPS in this regression when applied to YouTube: increasing the KNN view seems to be the only reliable settings to reach an accuracy of 1. However, Mignon aims at providing an accurate estimation with the smallest possible sizes for the KNN and RPS views, which makes it unreasonable to use a very large KNN to counteract the unreliability of RPS samples. We therefore conclude that Polynomial Least Squares regression cannot be successfully applied to a very steep distribution like the one in our YouTube dataset.

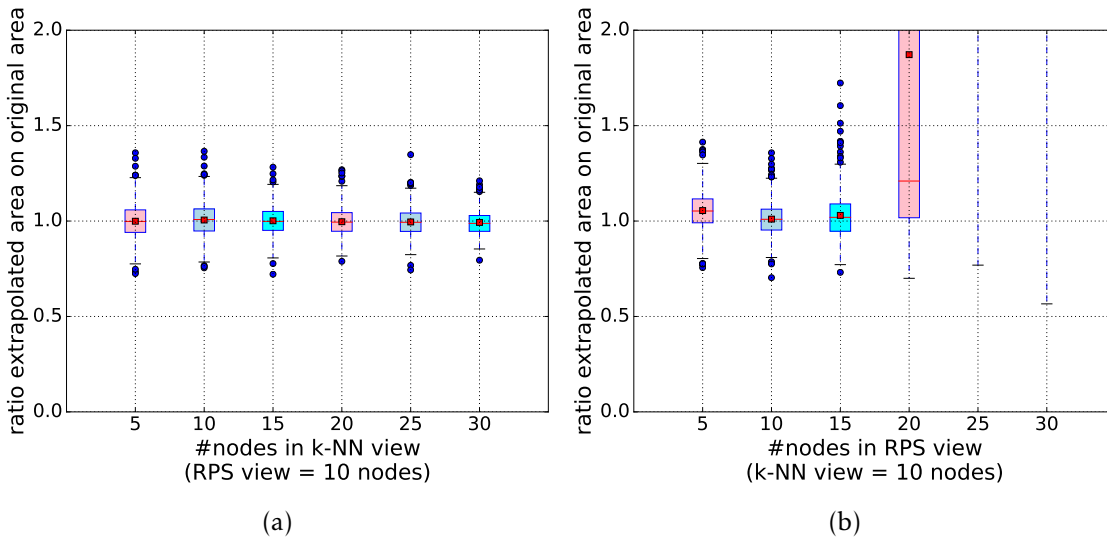


**Figure 5.14: Influence for the Polynomial Least Squares regression of varying the size of the KNN(a) and RPS(b) for the YouTube dataset.**



**Figure 5.15: Fast Decentralized Area Estimation using Gregory-Newton interpolation in the MovieLens dataset(a) and YouTube dataset(b).**

**Gregory-Newton interpolation.** Last, we evaluate the effectiveness of Mignon using polynomial interpolation. To this end, we used the Gregory-Newton interpolation forward difference algorithm as detailed in Equation 5.2. Figure 5.15 shows the results. Both datasets exhibit similar behaviours. For low RPS sizes, results resemble those obtained with the trapezoid rule, whether it be for MovieLens dataset or YouTube dataset, with the best performance being achieved with an RPS of 10 and a KNN of 15. However, results start diverging as soon as the RPS



**Figure 5.16: Influence for the Gregory-Newton interpolation of varying the size of the KNN(a) and RPS(b) for the MovieLens dataset.**

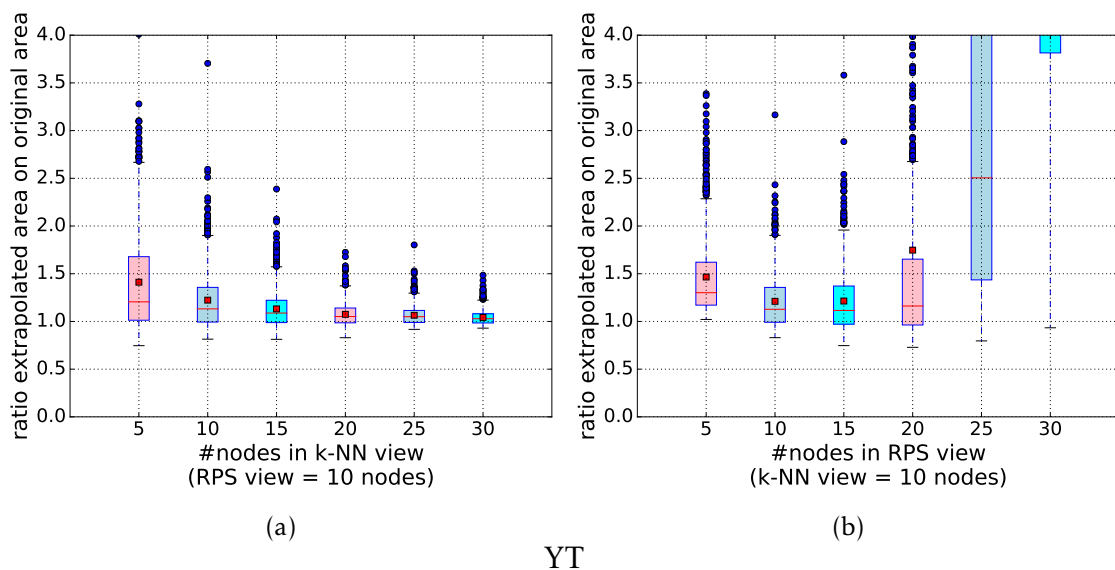
size goes beyond 15. We experimentally verified in Figure 5.16 and Figure 5.17 that this also occurs when increasing the RPS size with a constant KNN size, but not when increasing the KNN size with a constant RPS size. When we analyze the influence of varying the different settings for this method on a MovieLens dataset as Figure 5.16, the influence of the RPS view is pretty straight forward. When we fix the RPS size to 10, the influence of varying the KNN size seems nonexistent. As for the two firsts methods analyzed above, this method is not responsive to a variation of the KNN size, due to the shape of the MovieLens dataset. On the other hand we observe in Figure 5.16b the great influence of the RPS size on the area estimation. When it grows beyond a value of 15, the error ratios are quickly skyrocketing, along with their overall dispersion. The best values for the mean error and the standard deviation being reached as encountered before for a RPS size of 10.

When applied to a YouTube dataset, this sensitivity analysis reveals a more complex behavior, as show in Figure 5.17. In this case, both settings have their own and distinct influence on the accuracy of the results. In the first case, when we fix the RPS view and vary the KNN one, the mean error and the dispersion of the values improvement are coupled with the increase in KNN size. More we have nodes in the KNN view, the closer the mean and median ratios tend towards the ideal value of 1. The dispersion tends to shrink following a logarithmic shape when the KNN size increase. In our case, the best values are encountered for the

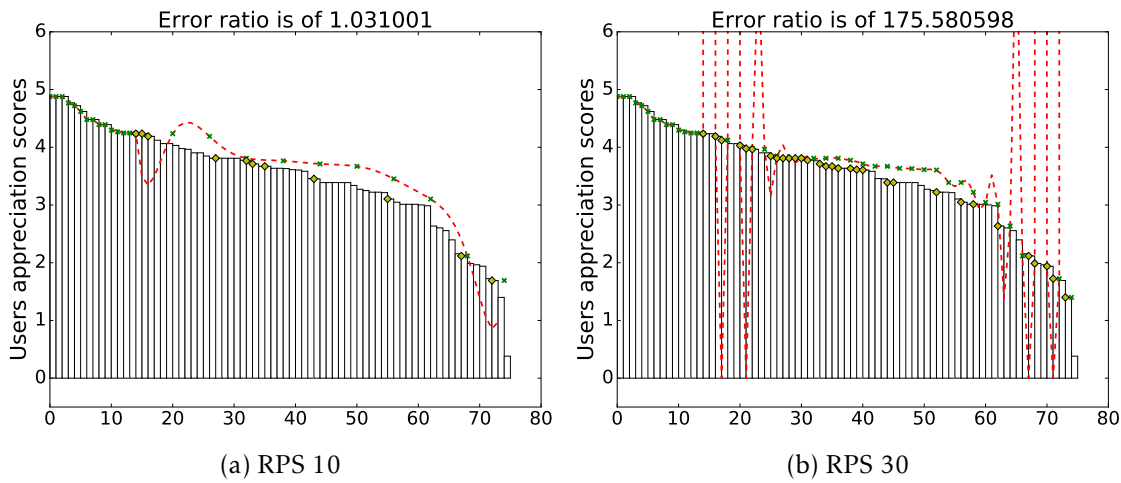
maximum KNN value of 30 reaching a mean error of the extrapolated ratio of 4.1% and a median one of 2.8%.

In the case of varying the RPS size we observe on the contrary that the increase of this settings cause a drastic increase in the error ratios distribution and their mean value. As encountered for the MovieLens case in Figure 5.16b, when we increase the value of RPS size beyond 15 the methods become less and less efficient, with a fast augmentation of the mean error and the dispersion of the estimation values. This overall behaviors when confronted to a distribution shaped like YouTube's one, can be explained by the method itself. Actually, the interpolation requires a constant spacing between the values serving the method, and as the YouTube distribution decrease very fast, it explains the need for a minimum size of the KNN view. On the other hand, the RPS size is directly responsible of the degree of the interpolated polynomial form, and the following analysis will demonstrate the instability of this method when confronted to a high RPS size.

To understand the high variability associated with high RPS sizes, we examine two runs of the Gregory-Newton interpolation algorithm in Figure 5.18. Figure 5.18a shows a run with 10 RPS nodes, while Figure 5.18b shows one with 30. In both figures, the diamonds represent the real abscissas of the samples on the curve, while the crosses represent those taken into account by our protocol (see Section 5.2). For KNN samples, the two coincide (points at the extreme left of the



**Figure 5.17: Influence for the Gregory-Newton interpolation of varying the size of the KNN(a) and RPS(b) for the YouTube dataset.**



**Figure 5.18: Details of the Gregory-Newton interpolation with different RPS sizes in the Movielens dataset.**

curve), but for the RPS the difference can be very large. This, together with the numerical instability of the Gregory-Newton’s method causes oscillations at the right end of the curve. Some oscillations are visible even with an RPS of 10, but with an RPS of 30, they completely disrupt the estimation.

## 5.4 Discussion

We proposed three different methods to estimate the aggregated interest of a country towards a content, with only morcelar informations. We analyzed the strengths and weaknesses when confronted to two different datasets, each one with it’s own shape of distribution.

The Trapezoidal rule and the Polynomial Least Squares regression can perform very well under certain conditions, each one having its own domain of predilection where to perform best. The third method, on the contrary appears as the best trade-off for the different situations. Able to perform relatively well in all the cases, this method represent our predilection choice among the others. As we want a fast and fully decentralized system, we cannot afford that each node successively try the three methods and pick the most adapted to the situation.

Moreover, as we noted that the biggest flaw of the Gregory-Newton interpolation is the instability brought by a large RPS view, bringing a high degree resolution, we are convinced that by working on gradually determining the best degree for the interpolation we can outperform the others methods. Even more,

Error	0%	+10%	-10%
MovieLens	-0.8%	+8.8%	-11%
YouTube	+12.4%	+14.9%	+8.7%

(a)

Error	0%	+10%	-10%
MovieLens	-0.6%	+8.9%	-11.1%
YouTube	+14.3%	+10.4%	+17%

(b)

**Table 5.1: Mean error percentage for various size-estimation errors, for Gregory-Newton interpolation(a) and Trapezoidal rule(b).**

as the Gregory-Newton forward difference is a sum where each element represent one degree of the polynomial, gradually increasing the number of degrees of the polynomial in order to determine which degree is the most fit for each case have exactly the same cost than calculating a  $n$  degree polynomial when we have an RPS view of  $n$  nodes, as we currently do.

#### 5.4.1 Influence of Sample & Collide

We now assess the impact of errors on the network-size estimation. As previously stated, nodes do not need to recompute the size of the network for every new upload as we assume the network to be relatively stable. Nonetheless, it is possible to limit the cost of size estimation by means of protocols like Sample & Collide [48]. Such a protocol yields an estimate with a 10% error at a very limited network cost. We estimate the impact of this error in Table 5.1 where we shows the absolute value of the error on the mean estimation ratio for the two most relevant Mignon’s approaches in the presence of a positive or negative error on the estimation size. The data shows that the error on the network size has almost no impact on YouTube, and a relatively low one on MovieLens.

#### 5.4.2 Convergence speed

We conclude by evaluating the time required to compute the estimate using Mignon. First, let us consider a baseline system that would simply compute the sum of the affinities of the uploaded video with all the other nodes in the country. Such a system would either require the uploading node to contact each other node in the country to compute its affinity, or it would have to disseminate the video’s profile so that other nodes could evaluate the video’s affinity with them. Both of

these approaches would clearly be difficult to scale to large numbers of nodes and their convergence time would be comparable, if not worse, than that required by a KNN protocol to converge from a completely random configuration.

Mignon, on the other hand, takes advantage of the presence of an already converged KNN protocol. This overlay allows the uploading node to quickly reach its closest neighbours. To evaluate this difference, we counted the number of gossip cycles required by a KNN protocol to reach convergence from scratch with 6000 nodes. In each cycle, a node contacts one other node, and is, on average, contacted by another one. We then added one random node, and counted the cycles it took to reach convergence again. Convergence from scratch took between 150 and 190 gossip cycles, while convergence after adding a node to an already converged network took an order of magnitude less (10 – 20).

## 5.5 Summary

In this Chapter we introduced different approaches to implement Mignon, a new protocol and an architecture to rapidly estimate the aggregate affinity of a newly uploaded video in a community of users, and place this video accordingly in a fully decentralized manner. Our proposal avoids an explicit and costly aggregation by relying on the properties of similarity-based self-organizing overlay networks, and can be used to decide where to place content in a decentralized UGC system. The core of this protocol resides in the ability to estimate the interest of the nodes present in a geographically limited area, in our case a country, with only partial informations obtained from the self-organizing overlay networks. We studied different approaches to solve this problem, and chose to use the Gregory-Newton interpolation for this role. Our architecture is now free of any central point, and allow any node in the system to determine when uploading a new content where in the world this content will be most consumed in the future.

**Part III**

**Conclusion**





## General Conclusion

This thesis aimed to tackle some open issues related to alternatives distributed storage systems. The goal was to investigate a solution aiming at performing predictive placement in a distributed User-Generated Content (UGC) service, in order to place the produced content close to where it is likely to be consumed in the future. The approach we proposed does not seek to compete with big placement and distribution systems operated by large tech companies, but rather to explore an alternative way applicable to distributed systems with a granularity of the scale of a personal computer or a set-top box. We firmly believe that an efficient predictive placement mechanism in a distributed UGC service that places the content within the geographic areas in which this content is more likely to be consumed helps in maximizing the share of overall local traffic, thus delivering an overall network that is more flexible and more apt to accomplish other tasks. We choose to take an unconventional approach, by studying metadata in large UGC systems like YouTube or MovieLens, especially tags as reliable consumption markers, because as far as we know, this way to tackle the problem has never been explored before. We further describe the three main obstacles and their respective solutions that allowed to present in this thesis out Architecture for Tag-Based Predictive Placement in Distributed Storage Systems.

### **Tags as reliable consumption markers**

The first challenge was to discover a reliable consumption marker, able to give us hints about the geographic distribution of views shares of a given content. Modern UGC contains a variety of metadata comprised of various markers.

Among them, we wanted to focus on user-generated metadata, in a front will to keep our system reliant on informations provided freely and voluntarily by the users producing the contents. As described in Chapter 3, our interest turned towards the tags, as this system of contextualisation of content is widely admitted and adopted, since its introduction in 1997 and its popularisation by *del.icio.us*<sup>1</sup> in 2007. We conducted our study by analysing a large dataset from YouTube, where users can freely attach tags to their uploaded videos. We first reconstructed the geographic distribution of videos views from the geographic popularity vectors. We analysed the distribution of video views via an entropy analysis of the distribution of views. We then performed the same analysis for each unique tag present in the dataset, and identified the presence of a correlation between the geographical distribution of views of a given video, and the ones of each unique tag embedded in the metadata of this video. This first part allowed us to confirm that tags could be used as reliable markers of the geographic distribution of views in UGC systems.

### Tag-based consumption prediction

The second challenge we needed to overcome related directly to the resolution of the first one. Having found a reliable non-intrusive marker allowing us to use it as a reflection of the geographic distribution of users' interests towards a content, we could now go further. We wanted to establish the viability of these markers to be used as a base to predict where a new video might be viewed. In short, we knew that tags could reflect the geographic spread of a video content, we then wanted to know if we could *predict* the future geographic dispersion of views for a given content, just by looking at the tags attached to it by the producer. We investigated this question in Chapter 4, by dividing the problem in two parts.

The first part aimed at establishing the extent to which we could rely on tags to predict the geographic distribution of a content. To do so, we split our dataset in two equal parts, both in terms of the number of videos as of the number of total views, and used one half as a *training set* to learn about the geographic distribution of the tags present inside. We then projected the videos from the second half — the *testing set* — on this knowledge, and with our metric, we evaluated the gap between the accuracy of prediction using our tag-based method and our baseline. The results highlighted that via a simple study and the basics manipu-

---

<sup>1</sup>Delicious.com

lations contained in our tag-based approach, we were able to predict the actual shares of views with a minimum of 65.9% of accuracy for half of our *testing set*, greatly outperforming our baseline.

Building upon these results, we attacked the second part of this challenge. We wanted to study how we could integrate this ability given by the tags, inside a proactive placement system, and have the possibility to design an efficient *proactive* placement system based on these results. To do so, we had to design a system where each user can produce and consume content. In this first simplified design, we assumed that the produced content is sent to a central entity where the attached tags are analyzed and confronted to the *training set*, before being placed to a local cache, one per country, made of limited storage and cache parts ruled by an eviction policy. By replaying the requests attached to videos in the *testing set*, we were able to determine if the consumers were able to find the requested content close to them in their local cache, or if it was a miss and their local cache had to fetch it from the central entity. Our metric measuring the hit/miss ratios between our approach and the baseline, confirmed the potential of our approach.

## **Distributed predictive placement system**

Our last challenge consisted of implementing our prediction and proactive placement approach in a decentralized environment. We had been able to rely on tags to perform an accurate proactive placement system, we now needed to perform these operations in a fast and scalable way inside a distributed storage system. We proposed two complementary approaches to tackle this problem.

The first one was to rely self-organising overlays, to organize peers inside a particular geographic area (in our case countries) according to the similarity of their previously consumed content. By allowing any node to reach at least one node inside each self-organized overlay, we addressed the problem of scalability and robustness. We used these per-country similarity overlays to compute an extrapolation of each country's interest in newly uploaded videos. This extrapolation uses a novel decentralized approach to estimate sums in large networks which we showed to be fast and efficient, while incurring little traffic as the video content is only moved as many times as the number of replicas. Taken together these different mechanisms pave the way for a practical predictive content placement solution for fully decentralized systems, based solely on the observation of the tags attached to videos.

## Perspectives

Although promising, our results do not provide a final answer to problem of designing efficient and scalable decentralized UGC services. In the following we review some of the future lines of research we think might be interesting to pursue in the light of the contributions presented in this thesis.

First, most of our analysis and our evaluation is based on a single dataset obtained from Youtube in 2011. Although extremely valuable, and of reasonable size, this dataset pales in comparison to the actual size of the information available to Youtube. The practice and behavior of uses might also have evolved since 2011, and it would therefore be interesting to confirm our findings on other, larger and more recent datasets, even though such detailed information regarding UGC services is particularly difficult to obtain.

In terms of actual evaluation, our results were obtained using simulation. Because of its versatility, and ease of use, simulation is particularly attractive to rapidly obtain useful results. Simulation does however abstract away many critical features of real systems, in particular when applied to distributed applications. A next step building on the presented work would therefore be to deploy some of the mechanisms we have presented in a real infrastructure to better understand their strengths and weaknesses in terms of performance and applicability.

More generally the work we have presented hints at the role that meta-data can play to improve the execution and performance of large-scale decentralized systems. This thesis has focused on a specific type of meta-data (*tags*), but a large variety of additional information could be used to make scheduling and resource allocation decisions beyond the mere placement of videos. In a similar vein, the approaches we have used for our predictions are rather basic (essentially average-based linear interpolations), and could be improved by exploiting more fully modern machine learning techniques, thus opening promising avenues for future research work on *intelligent decentralized systems*.

## Remerciements

Ce travail a vu le jour grâce à différentes participations formelles et informelles:

Merci à François TAÏANI et Davide FREY pour leur encadrement et leur excellente pédagogie durant ces trois années avec eux.

Merci également à la communauté du logiciel libre pour son fantastique travail, ainsi qu'à tous ceux qui cherchent des solutions complexes à des problèmes inexistantes.

Un merci à Fato, aux amis et à toutes les personnes rencontrées qui m'ont poussé à tenter cette aventure.

Un immense merci à ma famille pour leur présence et leur support continu à travers toutes mes aventures depuis de très nombreuses années.

Bon, depuis toujours en fait.

*“Et en ces âges farouches, merci à tous Ceux qui marchent Debout. -M.”*



# Bibliography

- [1] <http://www.youtube.com/yt/press/statistics.html>. (accessed 2/5/2014 and 10/10/2016).
- [2] <https://www.twitch.tv/p/about>. (accessed 10/10/2016).
- [3] [https://developers.google.com/chart/image/docs/gallery/map\\_charts](https://developers.google.com/chart/image/docs/gallery/map_charts).
- [4] <http://www.google.com/trends/>.
- [5] <http://www.alexa.com/>.
- [6] International telecommunication union. <http://www.itu.int>.
- [7] Global internet phenomena report: 2H 2012. Technical report, Sandvine Incorporated, 2012.
- [8] Global internet phenomena report: 2H 2013. Technical report, Sandvine Incorporated, 2013.
- [9] Global internet phenomena report: 2H 2015. Technical report, Sandvine Incorporated, 2015.
- [10] Twitch yearly retrospective. Technical report, Twitch.tv, 2015. <https://www.twitch.tv/year/2015>.
- [11] Netflix investors relations overview. Technical report, Netflix, 2016. <https://ir.netflix.com/>.
- [12] Chris Anderson. *The Long Tail: Why the Future of Business Is Selling Less of More*. Hyperion, 2006.



- [13] Stephanos Androutsellis-Theotokis and Diomidis Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Comput. Surv.*, 36(4):335–371, December 2004.
- [14] Xiao Bai, Marin Bertier, Rachid Guerraoui, Anne-Marie Kermarrec, and Vincent Leroy. Gossiping personalized queries. In *EDBT*, 2010.
- [15] Ranieri Baraglia, Patrizio Dazzi, Matteo Mordacchini, and Laura Ricci. A peer-to-peer recommender system for self-emerging user communities based on gossip overlays. *J. of Comp. and Sys. Sciences*, 2013.
- [16] Marin Bertier, Davide Frey, Rachid Guerraoui, Anne-Marie Kermarrec, and Vincent Leroy. The gossple anonymous social network. In *Middleware*, 2010.
- [17] A. Boutet, D. Frey, R. Guerraoui, A. Jégou, and A.-M. Kermarrec. WhatsUp Decentralized Instant News Recommender. In *IPDPS*, 2013.
- [18] Anders Brodersen, Salvatore Scellato, and Mirjam Wattenhofer. YouTube around the world: Geographic popularity of videos. In *WWW*, 2012.
- [19] Ripe Network Coordination Centre. Egyptian internet outage. <https://stat.ripe.net/events/egypt-outage>, January 2011.
- [20] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. I tube, you tube, everybody tubes: Analyzing the world’s largest user generated content video system. In *IMC*, 2007.
- [21] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Designing Privacy Enhancing Technologies*, pages 46–66. Springer, 2001.
- [22] Arturo Crespo and Hector Garcia-Molina. Routing indices for peer-to-peer systems. In *ICDCS*, 2002.
- [23] Stéphane Delbruel, Davide Frey, and François Taïani. Decentralized view prediction for global content placement. In *Proceedings of the 14th International Workshop on Adaptive and Reflective Middleware, ARM@Middleware 2015, Vancouver, BC, Canada, December 7-11, 2015*, pages 10:1–10:3. ACM, 2015.

- [24] Stéphane Delbruel, Davide Frey, and François Taïani. Exploring The Use of Tags for Georeplicated Content Placement. In *IEEE International Conference on Cloud Engineering IC2E'16*, Berlin, Germany, April 2016.
- [25] Stéphane Delbruel, Davide Frey, and François Taïani. Mignon: a Fast Decentralized Content Consumption Estimation in Large-Scale Distributed Systems. In *16th IFIP International Conference on Distributed Applications and Interoperable Systems DAIS'16*, Heraklion, Greece, June 2016.
- [26] Stéphane Delbruel and François Taïani. From views to tags distribution in youtube. In Romain Rouvoy, editor, *Proceedings of the Middleware '14 Posters & Demos Session, Bordeaux, France, December 8-12, 2014*, pages 21–22. ACM, 2014.
- [27] Stéphane Delbruel and François Taïani. Géodistribution des tags et des vues dans Youtube. In *Conférence d'informatique en Parallélisme, Architecture et Système. Compas'2015.*, Lille, France, June 2015.
- [28] M.D. Ekstrand, J.T. Riedl, and J.A. Konstan. *Collaborative Filtering Recommender Systems*. Now Publishers, 2011.
- [29] M. El Dick, E. Pacitti, and B. Kemme. Flower-cdn: a hybrid p2p overlay for efficient query processing in cdn. In *EDBT '09*, pages 427–438. ACM, 2009.
- [30] Manal El Dick, Esther Pacitti, and Bettina Kemme. Flower-cdn: A hybrid p2p overlay for efficient query processing in cdn. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, EDBT '09*, pages 427–438, New York, NY, USA, 2009. ACM.
- [31] Davide Frey, Mathieu Goessens, and Anne-Marie Kermarrec. Behave: Behavioral Cache for Web Content. In *DAIS*, 2014.
- [32] Davide Frey, Rachid Guerraoui, Anne-Marie Kermarrec, Boris Koldehofe, Martin Mogensen, Maxime Monod, and Vivien Quéma. Heterogeneous gossip. In *Middleware*. 2009.
- [33] Gary Geisler and Sam Burns. Tagging video: conventions and strategies of the youtube community. In *ACM/IEEE-CS Joint Conf. on Digital Libraries*, 2007.

- [34] Google Inc. Google peering and content delivery .  
<https://peering.google.com/about/ggc.html>. (accessed 2/5/2015).
- [35] S. Greenaway, M. Thelwall, and Y. Ding. Tagging youtube - a classification of tagging practice on youtube. In *Int. Conf. on Scientometrics and Informetrics*, 2009.
- [36] Krishna P Gummadi, Richard J Dunn, Stefan Saroiu, Steven D Gribble, Henry M Levy, and John Zahorjan. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. *ACM SIGOPS Operating Systems Review*, 37(5):314–329, 2003.
- [37] Martin Hilbert and Priscila López. The world’s technological capacity to store, communicate, and compute information. *science*, 332(6025):60–65, 2011.
- [38] Cheng Huang, Angela Wang, Jin Li, and Keith W. Ross. Understanding hybrid CDN-P2P: Why limelight needs its own red swoosh. In *NOSSDAV*, 2008.
- [39] Qi Huang, Ken Birman, Robbert van Renesse, Wyatt Lloyd, Sanjeev Kumar, and Harry C Li. An analysis of facebook photo caching. In *SOSP*, 2013.
- [40] Kévin Huguenin, Anne-Marie Kermarrec, Konstantinos Kloudas, and François Taïani. Content and geographical locality in user-generated content sharing systems. In *NOSSDAV*, 2012.
- [41] Stratos Idreos, Manolis Koubarakis, and Christos Tryfonopoulos. P2P-diet: An extensible P2P service that unifies ad-hoc and continuous querying in super-peer networks. In *SIGMOD*, pages 933–934. ACM, 2004.
- [42] Márk Jelasity, Alberto Montresor, and Ozalp Babaoglu. T-man: Gossip-based fast overlay topology construction. *Comput. Netw.*, 53(13):2321–2339, August 2009.
- [43] Márk Jelasity, Spyros Voulgaris, Rachid Guerraoui, Anne-Marie Kermarrec, and Maarten van Steen. Gossip-based peer sampling. *ACM TOCS*, 25, 2007.
- [44] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceeding of the 14th ACM SIGKDD international*

- conference on Knowledge discovery and data mining*, KDD '08, pages 426–434, New York, NY, USA, 2008. ACM.
- [45] John Kubiawicz, David Bindel, Yan Chen, Steven Czerwinski, Patrick Eaton, Dennis Geels, Ramakrishan Gummadi, Sean Rhea, Hakim Weatherpoon, Westley Weimer, et al. Oceanstore: An architecture for global-scale persistent storage. *ACM Sigplan Notices*, 35(11):190–201, 2000.
- [46] E. Le Merrer, A.-M. Kermarrec, and L. Massoulie. Peer to peer size estimation in large and dynamic networks: A comparative study. In *HPDC*, 2006.
- [47] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.
- [48] Laurent Massoulié, Erwan Le Merrer, Anne-Marie Kermarrec, and Ayalvadi Ganesh. Peer counting and sampling in overlay networks: Random walk methods. In *PODC*, 2006.
- [49] Ralph C. Merkle. *A Digital Signature Based on a Conventional Encryption Function*, pages 369–378. Springer Berlin Heidelberg, Berlin, Heidelberg, 1988.
- [50] A. Montresor, M. Jelasity, and O. Babaoglu. Robust aggregation protocols for large-scale overlay networks. In *DSN*, 2004.
- [51] Henry H Perritt. The internet as a threat to sovereignty? thoughts on the internet's role in strengthening national and global governance. *Indiana Journal of Global Legal Studies*, pages 423–442, 1998.
- [52] Marshall T Poe. *A History of Communications: Media and Society from the Evolution of Speech to the Internet*. Cambridge University Press, 2010.
- [53] J. M. Pujol, V. Erramilli, G. Siganos, X. Yang, N. Laoutaris, P. Chhabra, and P. Rodriguez. The little engine(s) that could: scaling online social networks. In *SIGCOMM*, 2010.
- [54] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. A scalable content-addressable network. In *SIGCOMM*, 2001.

- [55] Cloudflare Traffic Report. How syria turned off the internet. <https://blog.cloudflare.com/how-syria-turned-off-the-internet/>, November 2012.
- [56] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: An open architecture for collaborative filtering of netnews. In *CSCW*. ACM, 1994.
- [57] M. Ripeanu. [15] peer-to-peer architecture case study: Gnutella network. In *Proceedings of the First International Conference on Peer-to-Peer Computing, P2P '01*, pages 99–, Washington, DC, USA, 2001. IEEE Computer Society.
- [58] Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing*, pages 329–350. Springer, 2001.
- [59] Antony Rowstron and Peter Druschel. Storage management and caching in past, a large-scale, persistent peer-to-peer storage utility. In *Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles, SOSP '01*, pages 188–201, New York, NY, USA, 2001. ACM.
- [60] Stefan Saroiu, Krishna P Gummadi, and Steven D Gribble. Measuring and analyzing the characteristics of napster and gnutella hosts. *Multimedia systems*, 9(2):170–184, 2003.
- [61] Salvatore Scellato, Cecilia Mascolo, Mirco Musolesi, and Jon Crowcroft. Track globally, deliver locally: Improving content delivery networks by tracking geographic social cascades. In *WWW*, 2011.
- [62] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM '01*, pages 149–160, 2001.
- [63] Investigatory Powers Tribunal. Ipt/15/110/ch. [http://www.ipt-uk.com/docs/Bulk\\_Data\\_Judgment.pdf](http://www.ipt-uk.com/docs/Bulk_Data_Judgment.pdf), October 2016.
- [64] Shawn Wilkinson, Tome Boshevski, Josh Brandoff, and Vitalik Buterin. Storj a peer-to-peer cloud storage network, 2014.