

Ατομική Διπλωματική Εργασία

**Μελέτη Και Ανάλυση Της Επιρροής Του Κοινωνικού Δικτύου
Twitter Στο YouTube**

Χριστοδούλου Γεώργιος

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2012

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μελέτη και ανάλυση της επιρροής του κοινωνικού δικτύου twitter στο YouTube

Χριστοδούλου Γεώργιος

Επιβλέποντες Καθηγητές
Γεωργίου Χρύσης
Πάλλης Γιώργος

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2012

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τους επιβλέποντες καθηγητές μου Δρ Γιώργο Πάλλη και Δρ Χρύση Γεωργίου για την πολύτιμη βοήθεια και καθοδήγηση τους σε αυτή την διπλωματική εργασία.

Επίσης θα ήθελα να ευχαριστήσω τους συναδέλφους μου στο εργαστήριο διαδικτυακού υπολογισμού για τις συμβουλές τους και τον ωραίο και διασκεδαστικό χρόνο που περάσαμε μαζί.

Ένα μεγάλο ευχαριστώ στους φίλους και στην οικογένεια μου για την ψυχολογική τους στήριξη και κατανόηση για την έλλειψη χρόνου για αυτούς.

Περίληψη

Η έκρηξη της χρήσης του διαδικτύου που ήρθε με το νερχομό του Web 2.0 έχει δημιουργήσει νέα δεδομένα στον τρόπο με τον οποίο διαδίδεται και μοιράζεται η πληροφορία. Η έννοια της στατικής ιστοσελίδας που ανανεωνόταν μία φορά την εβδομάδα ανήκει στο μακρινό παρελθόν. Το διαδίκτυο πλέον κυριαρχείται από το user generated content, περιεχόμενο που δημιουργούν οι ίδιοι οι χρήστες για τους ίδιους τους χρήστες.

Αυτή η μετατόπιση του επίκεντρου από το περιεχόμενο στους χρήστες δημιουργεί νέα δεδομένα για τους μηχανισμούς διάδοσης του περιεχομένου στο διαδίκτυο. Πλέον το περιεχόμενο είναι πιο πολύ έχει μεγαλώσει σε μέγεθος και οι χρήστες είναι περισσότερο απαιτητικοί από ποτέ.

Στόχος της διπλωματικής αυτής είναι η αξιοποίηση της πληθώρας των πληροφοριών που υπάρχει για τους χρήστες στα κοινωνικά δίκτυα για να βελτιωθούν οι μηχανισμοί που χρησιμοποιούνταν για τη διανομή περιεχομένου μέχρι τώρα. Συγκεκριμένα το πώς οι σύνδεσμοι που μοιράζονται μεταξύ των χρηστών στα κοινωνικά δίκτυα μπορούν να χρησιμοποιηθούν για να μαντέψουν μελλοντικές αιτήσεις από τους χρήστες έτσι ώστε το περιεχόμενο να μεταφερθεί κοντά στους χρήστες μειώνοντας τον χρόνο καθυστέρησης και τη συμφόρηση του δικτύου.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
	1.1 Κίνητρα	1
	1.2 Στόχοι διπλωματικής εργασίας	2
	1.3 Μεθοδολογία που ακολουθήθηκε	2
	1.4 Οργάνωση Διπλωματικής εργασίας	3
Κεφάλαιο 2	Συναφής Βιβλιογραφία	3
	2.1 Κοινωνικά δίκτυα	4
	2.2 Το μοντέλο της διάδοσης στα κοινωνικά δίκτυα	4
	2.3 Συναφή συστήματα	8
Κεφάλαιο 3	Συλλογή δεδομένων	19
	3.1 Συλλογή κοινωνικών δεδομένων	19
	3.2 Εξαγωγή των πραγματικών συνδέσμων από τα Tweets	22
	3.3 Εύρεση της δημοτικότητας των βίντεο στο YouTube	23
	3.4 Εύρεση της τοποθεσίας των χρηστών	24
	3.5 Προβλήματα κατά τη συλλογή δεδομένων	24
	3.6 Σχηματική αναπαράσταση της Ιχνηλασίας των δεδομένων	25
Κεφάλαιο 4	Ανάλυση των διαδόσεων	26
	4.1 Συντελεστής διάδοσης	26
	4.2 Συντελεστής διάδοσης σχετικά με τις ζώνες ώρας	28
	4.3 Συντελεστής διάδοσης σχετικά με τον αριθμό των φίλων που διαδίδουν	29
	4.4 Καθυστέρηση διάδοσης	30
	4.5 Σχέση μεταξύ κοινωνικής διάδοσης και δημοτικότητας	31
Κεφάλαιο 5	Γεννήτρια αιτήσεων για το YouTube	33
	5.1 Εισαγωγή	33
	5.2 Κοινωνικές πληροφορίες	33
	5.3 Πληροφορίες αιτήσεων	34
	5.4 Αλγόριθμος	36

Κεφάλαιο 6	Εφαρμογές	40
	6.1 Προβλήματα με τα δίκτυα παράδοσης περιεχομένου	40
	6.2 Πολυμέσα	41
	6.3 Κοινωνικά δίκτυα	42
	6.4 Διατύπωση προβλήματος	44
Κεφάλαιο 7	Συμπεράσματα	46
	7.1 Συμπεράσματα	46
	7.2 Μελλοντική εργασία	46
Βιβλιογραφία		48
Παράρτημα Α.....		Α-1
Παράρτημα Β.....		Β-1
Παράρτημα Γ.....		Γ-1

Λίστα σχημάτων

Σχήμα 2.1 Αύξηση δημοτικότητας στο Flickr [2]	5
Σχήμα 2.2 Αθροιστική συνάρτηση κατανομής του αριθμού των πιθανών μολυντών [2]	7
Σχήμα 2.3 Υπολογισμός του αριθμού αναπαραγωγής R_0 [2]	8

Σχήμα 2.5 Σύγκριση κόστους μεταξύ τοποθέτησης με βάση τη θέση και κοινωνικής τοποθέτησης.....	12
Σχήμα 2.6 Μέσο κόστος για διάφορες τιμές του Ps [3].....	12
Σχήμα 2.7 Αθροιστική συνάρτηση κατανομής της τοπικότητας των κόμβων [4]	13
Σχήμα 2.8 Συνάρτηση αθροιστικής κατανομής του χρόνου καθυστέρησης μεταξύ 2 συνεχόμενων tweets και τη συνολική διάρκεια διάδοσης [4]	14
Σχήμα 2.9 Georange [4]	15
Σχήμα 3.1 Σύστημα ιχνηλασίας δεδομένων	25
Σχήμα 4.4 Συντελεστής διάδοσης σε σχέση με το μέγεθος των ζωνών ώρας σε λογαριθμική κλίμακα.....	29
Σχήμα 4.5 Συντελεστής διάδοσης σε σχέση με τον αριθμό των φίλων που μοιράστηκαν το σύνδεσμο	29
Σχήμα 4.6 Ποσοστό των βημάτων διάδοσης που έχουν γίνει ανάλογα με τις ώρες που έχουν περάσει	30
Σχήμα 5.1 Κατανομή των αιτήσεων στις ώρες της μέρας [8]	35
Σχήμα 5.2 Κατανομή των αιτήσεων στις διάφορες μέρες της εβδομάδας[8]	35

Λίστα Πινάκων

Πίνακας 2.4 Γεωγραφική κατανομή των συστάδων εξυπηρετητών στο ΔΠΠ Limelight [4]...	9
Πίνακας 4.1 Συντελεστής διάδοσης για ολόκληρο το χρονικό διάστημα στο οποίο έχουμε tweets.....	26
Πίνακας 4.2 Συντελεστής διάδοσης για 1 ^η Ιανουαρίου του 2012 μέχρι το τέλος Μαρτίου ...	27
Πίνακας 4.3 Συντελεστής διάδοσης για την εβδομάδα με τις περισσότερες δημοσιεύσεις....	27
Πίνακας 4.7 Κοινωνικές διαδόσεις σε σχέση με τον αριθμό των θεάσεων	32

Κεφάλαιο 1

Εισαγωγή

1.1 Κίνητρα	1
1.2 Στόχοι - Συνεισφορά διπλωματικής εργασίας	2
1.3 Μεθοδολογία Που Ακολουθήθηκε	2
1.4 Οργάνωση Διπλωματικής Εργασίας	3

1.1 Κίνητρα

Αυτές τις μέρες τα διαδικτυακά αντικείμενα μεγάλου μεγέθους (heavy bandwidth web objects) γίνονται ολοένα και πιο δημοφιλή. Ένας τρόπος για να κρατηθούν το κόστος και ο χρόνος απόκρισης (response time χαμηλά) είναι τα Δίκτυα Παράδοσης Περιεχομένου (Content Delivery Networks) [1]. Ένα δίκτυο παράδοσης περιεχομένου είναι ένα παγκόσμιο σύστημα προσωρινής αποθήκευσης (worldwide caching system) το οποίο φέρνει τα αντικείμενα πιο κοντά στο χρήστη αποθηκεύοντας τα σε εξυπηρετητές οι οποίοι είναι κοντά στους χρήστες.

Λόγω της πληθώρας των αντικειμένων που υπάρχουν στο διαδίκτυο και τις μεγάλες διαφορές που έχουν σε δημοτικότητα τα δίκτυα παράδοσης περιεχομένου υ δεν μπορούν να αντεπεξέλθουν το ίδιο καλά όπως σε προηγούμενα χρόνια. Το περιεχόμενο πλέον δημιουργείται από τους χρήστες και είναι άφθονο και διαφορετικό. Για να παρακολουθήσει κάποιος όλα τα βίντεο στο YouTube χρειάζεται 6,000 χρόνια ενώ στο YouTube ανεβάζονται κάθε μήνα βίντεο που έχουν όση διάρκεια έχουν όλες οι ταινίες που βρίσκονται στη βάση δεδομένων του IMDB (Internet Movie Database) [5].

Τα κοινωνικά δίκτυα έχουν φτάσει σε πρωτοφανή επίπεδα επισκεψιμότητας. Το κοινωνικό δίκτυο Facebook αναμένεται να φτάσει το ένα δισεκατομμύριο εγγεγραμμένους χρήστες μέχρι τον Αύγουστο του 2012 ενώ έχει 300 εκατομμύρια ενεργούς χρήστες κάθε μήνα¹. Ακολουθεί το Twitter με 500 εκατομμύρια εγγεγραμμένους χρήστες².

¹www.facebook.com/press/info.php?statistics

²<http://en.wikipedia.org/wiki/Twitter>

Η μεγάλη ανάπτυξη των κοινωνικών δικτύων δίνει ένα τεράστιο όγκο χρήσιμων πληροφοριών. 400 tweets κάθε λεπτό περιέχουν ένα σύνδεσμο στη σελίδα του YouTube. Τα κοινωνικά δίκτυα οδηγούν την κίνηση στο διαδίκτυο και μπορούμε να προβλέψουμε μέσω αυτών τι θα ζητήσουν οι χρήστες [6].

Χρησιμοποιώντας τις προτιμήσεις των χρηστών, την τοποθεσία τους και την κοινωνική τους δραστηριότητα μπορούμε να προβλέψουμε πιο περιεχόμενο θα γίνει δημοφιλές και που θα γίνει δημοφιλές έτσι ώστε να το αναπαράγουμε στις περιοχές που θα γίνει δημοφιλές μειώνοντας τον χρόνο απόκρισης για τους χρήστες που ζητούν αυτά τα αντικείμενα. Χρησιμοποιώντας τοπική αντιγραφή αντί καθολική ελαχιστοποιούμε το κόστος και επιτρέπουμε σε περισσότερα αντικείμενα να αντιγραφούν.

1.2 Στόχοι - Συνεισφορά Διπλωματικής Εργασίας

Η διπλωματική αυτή έχει ως στόχο να αναλύσει τις πληροφορίες που μπορούν να εξαχθούν από τα κοινωνικά δίκτυα και να θεσπιστεί μία βάση έτσι ώστε να μπορούν αυτές οι πληροφορίες να χρησιμοποιηθούν από τα Δίκτυα Παράδοσης Περιεχομένου.

Γίνεται ανάλυση της διάδοσης στα κοινωνικά δίκτυα και από ποιους παράγοντες επηρεάζεται. Αυτή εξ όσον γνωρίζουμε είναι η πρώτη εκτενής και εις βάθος ανάλυση των κοινωνικών διαδόσεων το φ και το υ αντίκτυπου τους σε ιστοσελίδες πολυμέσων, συγκεκριμένα της ιστοσελίδας YouTube.

Χρησιμοποιώντας τις πληροφορίες για τα κοινωνικά δίκτυα και το YouTube δημιουργείται μία γεννήτρια αιτήσεων η οποία μοντελοποιεί τις αιτήσεις στο YouTube χρησιμοποιώντας πραγματικές κατανομές και κοινωνικά δεδομένα. Αυτή η γεννήτρια αιτήσεων είναι απαραίτητη για τη δοκιμή και τον έλεγχο των συστημάτων τα οποία χρησιμοποιούν κοινωνικά δεδομένα για να βελτιώσουν την προσωρινή αποθήκευση περιεχομένου.

Επίσης αναλύεται η συναφής βιβλιογραφία και θεσπίζεται η βάση για τη χρησιμοποίηση των κοινωνικών δεδομένων για βελτίωση των Δικτύων Παράδοσης Περιεχομένου.

1.3 Μεθοδολογία Που Ακολουθήθηκε

Αρχικά αναλύεται η σχετική βιβλιογραφία σχετικά με τα δίκτυα παράδοσης περιεχομένου, τις υπηρεσίες που προσφέρουν πολυμεσικό περιεχόμενο και σχετικά συστήματα βρίσκοντας μειονεκτήματα και ελλείψεις και προβλήματα που χρειάζεται να επιλυθούν.

Ακολούθως γίνεται η διατύπωση του βασικού προβλήματος.

Ακολούθως γίνεται η συλλογή δεδομένων και η ανάλυση τους.

Γίνεται μία εκτενής ανάλυση των κοινωνικών διαδόσεων και των ιδιοτήτων τους και παρουσιάζεται η συσχέτιση με των διαδόσεων ενός αντικειμένου με την πραγματική του δημοτικότητα.

Παρουσιάζεται μία γεννήτρια αιτήσεων για την πλατφόρμα YouTube η οποία μοντελοποιεί τις αιτήσεις των χρηστών στην υπηρεσία.

Τέλος παρουσιάζονται τα συμπεράσματα και η μελλοντική εργασία.

1.4 Οργάνωση Διπλωματικής Εργασίας

Στο κεφάλαιο 2 περιγράφεται η συναφής βιβλιογραφία και παρουσιάζονται σχετικά συστήματα. Στο κεφάλαιο 3 περιγράφεται ο τρόπος συλλογής δεδομένων, τα δεδομένα που συλλεχτήκαν και τα προβλήματα που δημιουργήθηκαν. Στο κεφάλαιο 4 αναλύεται η διάδοση στο κοινωνικό δίκτυο Twitter και οι παράγοντες οι οποίοι επηρεάζουν την κοινωνική δικτύωση. Στο κεφάλαιο 5 περιγράφεται και αναλύεται η γεννήτρια αιτήσεων για τη σελίδα YouTube. Στο κεφάλαιο 6 περιγράφονται οι εφαρμογές. Στο κεφάλαιο 7 συζητούνται τα συμπεράσματα και η μελλοντική εργασία.

Κεφάλαιο 2

Συναφής βιβλιογραφία

2.1 Κοινωνικά δίκτυα	4
2.2 Το μοντέλο της διάδοσης στα κοινωνικά δίκτυα	4
2.3 Συναφή συστήματα	8

Παρουσιάζεται η σχετική βιβλιογραφία σχετικά με κοινωνικά δίκτυα, Δίκτυα Παράδοσης Περιεχομένου και συναφή συστήματα.

2.1 Κοινωνικά Δίκτυα

Τα κοινωνικά δίκτυα είναι σελίδες στις οποίες οι χρήστες επικοινωνούν και μοιράζονται πληροφορίες [6]. Το μεγαλύτερο κοινωνικό δίκτυο είναι το Facebook ακολουθούμενο από το Twitter. Στο Facebook υπάρχει η έννοια της κοινωνικής ροής (social feed) όπου εμφανίζονται οι δημοσιεύσεις, φωτογραφίες των φίλων του χρήστη στις οποίες ο χρήστης μπορεί να σχολιάσει και να δηλώσει ότι του αρέσουν. Το κοινωνικό δίκτυο του Twitter έχει εισάγει τον όρο Tweet (Τιτίβισμα) όπου ένα tweet είναι ένα μήνυμα 140 χαρακτήρων στο οποίο ο χρήστης ενημερώνει τους ακολούθους του για κάτι, από προσωπικά θέματα μέχρι πολιτικά. Το twitter είναι πλέον ένα κοινωνικό μέσο μαζικής ενημέρωσης. Αυτό φανερώνει και η διαφορά που έχει από το facebook και δεν υπάρχει η έννοια του φίλου αλλά του ακολούθου και επίσης ο γράφος δεν είναι κατευθυνόμενος.

2.3 Το Μοντέλο της Διάδοσης στα Κοινωνικά Δίκτυα

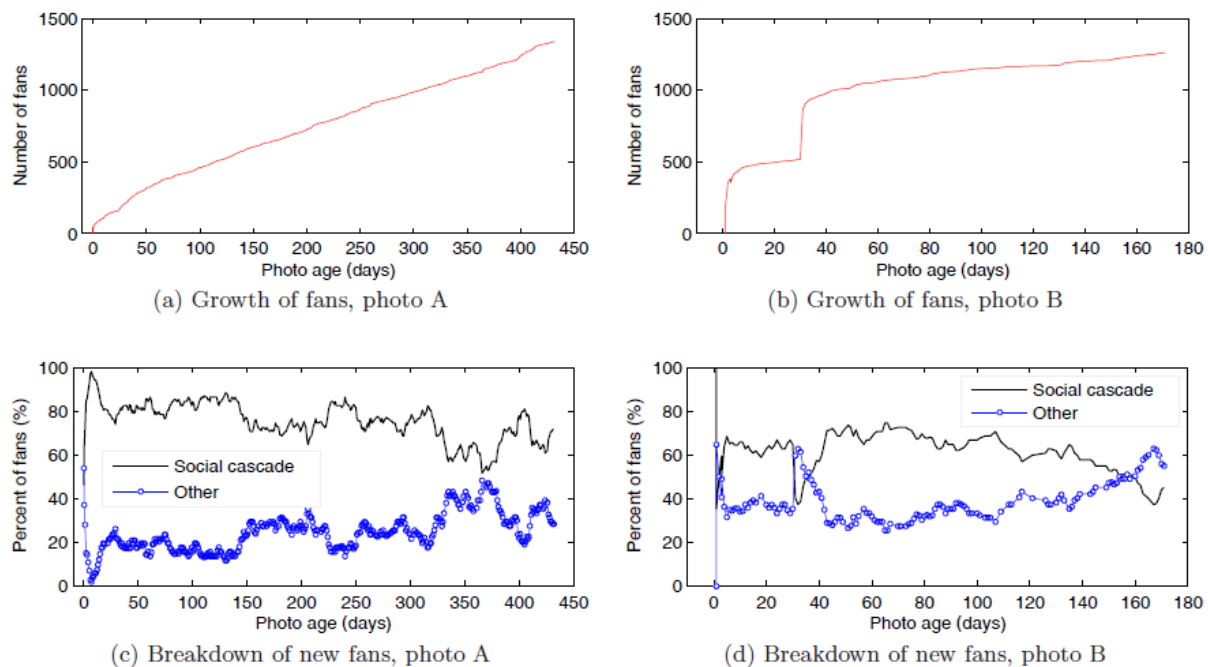
Μία μέθοδος για να ανακαλυφθούν τάσεις στη δημοτικότητα των αντικειμένων είναι η χρησιμοποίηση των κοινωνικών διαδόσεων (Social Cascades) [2]. Μια κοινωνική διάδοση ορίζεται ως η διάδοση περιεχομένου μεταξύ δύο ή περισσότερων χρηστών οι οποίοι σχετίζονται μέσα από το κοινωνικό δίκτυο. Για παράδειγμα έστω ότι υπάρχουν 2 χρήστες A και B οι οποίοι είναι φίλοι στο κοινωνικό δίκτυο. Ο χρήστης A μοιράζεται ένα βίντεο στην κοινωνική του ροή (social feed) και σε μεταγενέστερο χρόνο ο χρήστης B μοιράζεται τον

ίδιο σύνδεσμο στη κοινωνική του ροή. Τότε έχει γίνει μια κοινωνική διάδοση αυτού του συνδέσμου από τον χρήστη A στον χρήστη B.

Στο [2] γίνεται ανάλυση των κοινωνικών διαδόσεων στο κοινωνικό δίκτυο Flickr¹. Το Flickr είναι μια ιστοσελίδα κοινωνικής δικτύωσης που επιτρέπει στους χρήστες να ανεβάζουν φωτογραφίες έτσι ώστε οι φίλοι τους να μπορούν να τις δουν και να τις προσθέσουν στα αγαπημένα τους και έτσι οι φίλοι των φίλων τους θα μπορούν να την δουν και να κάνουν το ίδιο.

Για την ανάλυση είχε γίνει ιχνηλάτηση (crawling) του κοινωνικού δικτύου Flickr για 100 μέρες. Έχουν συλλεχθεί δεδομένα για 2,5 εκατομμύρια χρήστες και 33 εκατομμύρια ακμές μεταξύ τους. Έχουν ανευρεθεί 34 εκατομμύρια αγαπημένες φωτογραφίες, η στατιστική ανάλυση γίνεται στις 1180 δημοφιλή φωτογραφίες οι οποίες είχαν μαρκαριστεί ως αγαπημένες 171,131 φορές.

Με τη μελέτη της αύξησης των οπαδών και λαμβάνοντας υπόψη ότι η φωτογραφία διαδίδεται μέσω του κοινωνικού δικτύου, εφόσον ο ορισμός της κοινωνικής διάδοσης που περιγράφηκε πιο πάνω ισχύει για κάθε νέο οπαδό της εικόνας, το κλάσμα των οπαδών που είχαν βρει την εικόνα μέσω του κοινωνικού δικτύου εκτιμάται ότι είναι 40%.



Σχήμα 2.1 Αύξηση δημοτικότητας στο Flickr [2]

Για 2 φωτογραφίες βλέπουμε την ανάπτυξη των οπαδών και το ποσοστό των οπαδών, που βρήκε την εικόνα μέσα από την κοινωνική διάδοση και το ποσοστό που τη βρήκε με άλλα

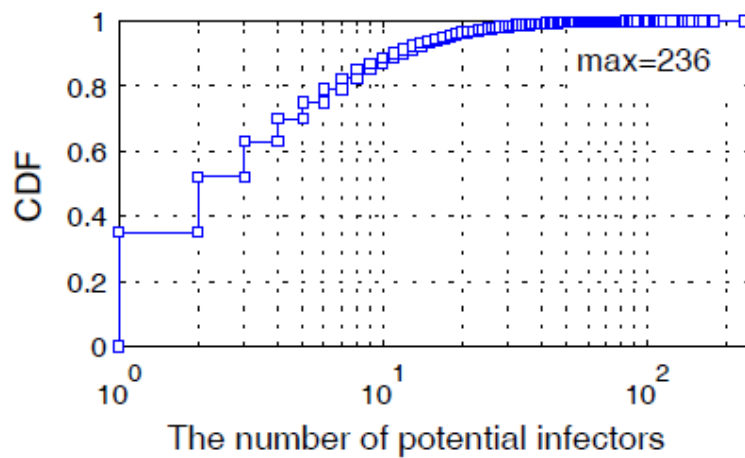
μέσα. Η πλειοψηφία των οπαδών βρήκε τις εικόνες μέσω της κοινωνικής διάδοσης αλλά στην φωτογραφία η Β απότομη αύξηση της δημοτικότητας της, δεν δημιουργήθηκε από την κοινωνική διάδοση το οποίο δείχνει ότι άλλα μέσα διάδοσης, όπως οι εξωτερικοί σύνδεσμοι και η εμφάνιση σε περίοπτες θέσεις στις διάφορες σελίδες (featuring) μπορούν να επηρεάσουν ένα μεγαλύτερο ποσοστό από τους χρήστες σε συντομότερο χρονικό διάστημα από την κοινωνική διάδοση.

Για την περεταίρω ανάλυση των χαρακτηριστικών των κοινωνικών διαδόσεων έγινε μια ανάλυση των 1000 πιο δημοφιλή φωτογραφιών. Υπάρχει ένα σύνολο φωτογραφιών J και ένα σύνολο χρηστών U . Κάθε χρήστης μπορεί να είναι σε μία από τις τρεις καταστάσεις. Αρχικά όλοι οι χρήστες είναι επιρρεπείς σε κάθε φωτογραφία (S_j^u). Όταν ένας χρήστης εκτίθεται σε μια φωτογραφία περνά σε μία από δύο καταστάσεις. Είτε ο χρήστης μολύνεται και προσθέτει τη φωτογραφία στα αγαπημένα του (I_j^u) ή αφαιρείται από την κοινωνική διάδοση (R_j^u). Το φαινόμενο ότι οι χρήστες δεν ανακάμψουν από τη μολυσμένη κατάσταση παρατηρείται επειδή οι χρήστες δεν αφαιρούν μια εικόνα από τα αγαπημένα του. Για να αφαιρεθεί ένας χρήστης από τη μολυσμένη κατάσταση το γεγονός ότι πιο πρόσφατες φωτογραφίες εμφανίζονται πρώτα στη λίστα με τις αγαπημένες του λαμβάνεται υπόψη και παλαιότερες φωτογραφίες δεν είναι πλέον ορατές, ώστε ο χρήστης μπορεί να αφαιρεθεί και από τη μολυσμένη κατάσταση. Αυτό ισχύει για όλα τα κοινωνικά δίκτυα, δεδομένου ότι δείχνουν όλα τα πιο πρόσφατα μηνύματα και δημοσιεύσεις χωρίς καμία δραστηριότητα για ένα χρονικό διάστημα είναι κρυμμένες στο κάτω μέρος της σελίδας δεν εμφανίζονται καθόλου.

Επίσης, οι συγγραφείς υπαινίσσονται ότι η συσχέτιση πρέπει να παρατηρηθεί μεταξύ του χρόνου λοίμωξης και το βαθμό του κόμβου. Οι κόμβοι με υψηλό βαθμό αναμένεται να αλλάζουν τα αγαπημένα τους πιο συχνά από ό, τι κόμβοι με χαμηλότερο βαθμό και έχουν ως εκ τούτου λιγότερη διάρκεια μόλυνσης από λιγότερο συνδεδεμένους χρήστες, που συνεπάγεται μείωση της μετάδοσης.

Παρατηρείται ότι 50% των πρώτων βημάτων διάδοσης γίνονται σε λιγότερο από 3 μέρες αλλά 20% παίρνουν περισσότερο από ένα μήνα. Ο χρόνος από τη στιγμή που ένας χρήστης εκτίθεται σε μια φωτογραφία μέχρι να την κατατάξει στις αγαπημένες του, στο 50% των

περιπτώσεων γίνεται εντός 5 0 ημερών ενώ κάποιες φορές παίρνει και χρόνια. 35% των διαδόσεων προκαλείται από ένα χρήστη, το 2 0% από 2 και το υπόλοιπο 4 9% από 3 ή περισσότερους.



Σχήμα 2.2 Αθροιστική συνάρτηση κατανομής του αριθμού των πιθανών μολυντών [2]

Για να κατανοήσουμε τα στατιστικά σχετικά με την κοινωνική διάδοση που παρουσιάστηκαν πρέπει να λάβουμε υπόψη ότι τα δεδομένα συλλέχθηκαν το 2008 και ότι το Flickr είναι ένας διαφορετικός τύπος Κοινωνικού δικτύου από τα συνήθη κοινωνικά δίκτυα όπως το Facebook και το Twitter. Ο μικρός αριθμός των χρηστών οι οποίοι διαδίδουν περιεχόμενο (infestors) που χρειάζονται για να γίνει η διάδοση δείχνει ότι η κοινωνική διάδοση διαδραματίζει πολύ σημαντικό ρόλο στη διάδοση του περιεχομένου.

Η κοινωνική διάδοση μπορεί να μοντελοποιηθεί χρησιμοποιώντας το επιδημιολογικό μοντέλο παραλληλίζοντας την εξάπλωση ενός αντικειμένου μεταξύ των χρηστών με τη μόλυνση ενός ιού από άνθρωπο σε άνθρωπο. Στο επιδημιολογικό μοντέλο R_0 ορίζεται ως ο αναμενόμενος αριθμός των δευτερογενών λοιμώξεων που προκαλούνται από ένα μόνο μολυσμένο άτομο σε ένα εντελώς ευπαθές πληθυσμό. Αν το R_0 είναι μεγαλύτερο από 1 η επιδημία εξαπλώνεται, όταν το R_0 είναι μικρότερο από 1 η επιδημία περιορίζεται και όταν το R_0 ισούται με 1 είναι το επιδημιακό κατώφλι όπου ο αριθμός των χρηστών που είναι μέρος της επιδημίας παραμένει σταθερός.

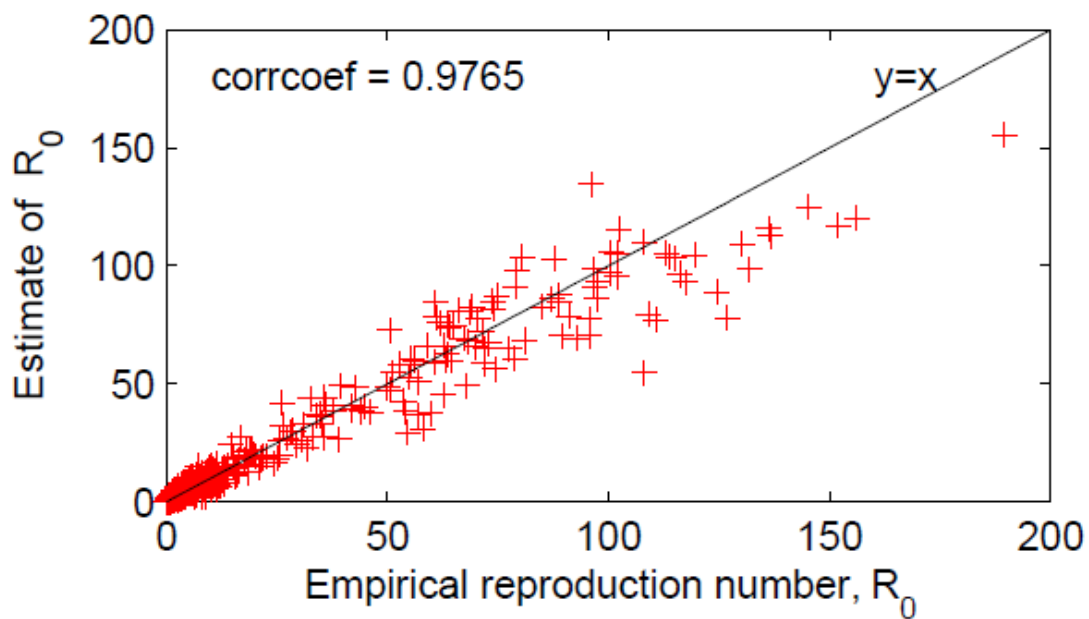
$$R_0 = \rho_0 \frac{\langle \kappa^2 \rangle}{\langle \kappa \rangle^2} \quad \text{όπου } \rho_0 = \beta \gamma \langle \kappa \rangle$$

β ο ρυθμός μετάδοσης

γ η διάρκεια της μόλυνσης

κ ο βαθμός του κόμβου

και $\langle \dots \rangle$ υποδηλώνει μέση τιμή.



Σχήμα 2.3 Υπολογισμός του αριθμού αναπαραγωγής R_0 [2]

2.3 Παρόμοια Συστήματα

2.3.1 Δίκτυα Παράδοσης Περιεχομένου

Τα Δίκτυα Παράδοσης Περιεχομένου (ΔΠΠ) [1] είναι ένας μεγάλος παράγοντας ο οποίος συνάδει στην ομαλή λειτουργία του διαδικτύου. Τα δίκτυα παράδοσης περιεχομένου λειτουργούν αντιγράφοντας περιεχόμενο σε διάφορες τοποθεσίες έτσι ώστε οι χρήστες να αντλούν περιεχόμενο από πιο κοντινές σε αυτούς τοποθεσίες μειώνοντας το χρόνο απόκρισης και τη συμφόρηση του δικτύου.

Τα δίκτυα παράδοσης περιεχομένου χρησιμοποιούν υποκατάστατους εξυπηρετητές (surrogate servers). Τα αντικείμενα του εξυπηρετητή προέλευσης (origin server) αντιγράφονται στους υποκατάστατους εξυπηρετητές. Οι χρήστες ζητούν αντικείμενα από τους υποκατάστατους εξυπηρετητές αντί από τον εξυπηρετητή προέλευσης. Έτσι μειώνεται το κόστος για τους ιδιοκτήτες του περιεχομένου και μειώνεται ο φόρτος εργασίας στους εξυπηρετητές προέλευσης. Ένα δίκτυο παράδοσης περιεχομένου είναι ένα παγκόσμιο σύστημα τοπικής αποθήκευσης.

Location	Country	Servers	Location	Country	Servers
Washington	USA	552	Frankfurt	Germany	314
Los Angeles	USA	523	London	UK	300
New York	USA	438	Amsterdam	Netherlands	199
Chicago	USA	374	Tokyo	Japan	126
San Jose	USA	372	Toronto	Canada	121
Dallas	USA	195	Paris	France	120
Seattle	USA	151	Hong Kong	Hong Kong	83
Atlanta	USA	111	Changi	Singapore	53
Miami	USA	111	Sydney	Australia	1
Phoenix	USA	3			

Πίνακας 2.4 Γεωγραφική κατανομή των συστάδων εξυπηρετητών στο ΔΠΠ Limelight [4]

Δύο είναι οι παράγοντες που επηρεάζουν την απόδοση των δικτύων παράδοσης περιεχομένου. Ο πρώτος είναι το που θα τοποθετηθούν οι υποκατάστατοι εξυπηρετητές. Μία καλή τοποθέτηση μειώνει το χρόνο απόκρισης και το κόστος εφόσον θα τοποθετηθούν πιο λίγοι εξυπηρετητές. Η τοποθέτηση πρέπει να λαμβάνει υπόψη την τοποθεσία των χρηστών, την τοποθεσία και το φόρτο εργασίας των εξυπηρετητών προέλευσης και την τοπολογία του δικτύου. Ο δεύτερος παράγοντας που επηρεάζει είναι το ποια αντικείμενα θα τοποθετηθούν σε ποιούς εξυπηρετητές. Κοινή στρατηγική είναι να αντιγράφεται το περιεχόμενο που ζητείται πιο συχνά από τους χρήστες. Το περιεχόμενο επιλέγεται αναλύοντας την δραστηριότητα των χρηστών ή και αναλύοντας την τοπολογία των ιστοσελίδων.

Μία αποτελεσματική μέθοδος για επιλογή περιεχομένου προς αντιγραφή είναι η εύρεση διαδικτυακών κοινοτήτων (Web Communities). Ο ορισμός μίας διαδικτυακής κοινότητας ορίζει ότι το άθροισμα των ακμών μέσα σε μια κοινότητα είναι μεγαλύτερος από το άθροισμα των ακμών που δείχνουν προς τα έξω. Άρα οι χρήστες είναι πολύ πιθανό εάν επισκεφθούν ένα κόμβο μιας κοινότητας να επισκεφθούν και άλλους κόμβους στην ίδια κοινότητα. Έτσι εντοπίζονται οι κοινότητες και αντιγράφονται ολόκληρες στους υποκατάστατους εξυπηρετητές.

Στρατηγικές αντιγραφής που εφαρμόζονται στα δίκτυα παράδοσης περιεχομένου:

Συνεργατικό με ώθηση (Cooperative push based): Το περιεχόμενο αντιγράφεται πριν ζητηθεί (pushed). Οι υποκατάστατοι εξυπηρετητές δουλεύουν μαζί. Το ΔΠΠ παρακολουθεί ποια αντικείμενα υπάρχουν στον κάθε εξυπηρετητή και επιλέγει τον κοντινότερο εξυπηρετητή στο

χρήστη που περιέχει το αντικείμενο για να παραδώσει το αντικείμενο στο χρήστη αλλιώς αν δεν υπάρχει η αίτηση μεταβιβάζεται στον εξυπηρετητή προέλευσης.

Μη συνεργατικό με έλξη (Uncooperative pull-based): Οι χρήστες ανακατευθύνονται στον πιο κοντινό υποκατάστατο εξυπηρετητή. Αν τον αντικείμενο δεν υπάρχει τότε μεταφέρεται από τον εξυπηρετητή προέλευσης(έλκεται, pulled). Δεν είναι και το καλύτερο μοντέλο εφόσον οι υποκατάστατοι εξυπηρετητές δεν επιλέγονται με τον καλύτερο τρόπο.

Συνεργατικό με έλξη (Cooperative pull-based): Οι χρήστες ανακατευθύνονται στον πιο κοντινό υποκατάστατο εξυπηρετητή και οι υποκατάστατοι εξυπηρετητές συνεργάζονται μεταφέροντας τα αντικείμενα ο ένας από τον άλλο.

Οι στρατηγικές που χρησιμοποιούν έλξη χρησιμοποιούν URL rewriting mechanisms και DNS redirection ενώ οι στρατηγικές που χρησιμοποιούν ώθηση χρησιμοποιούν ένα κατανεμημένο δείκτη (distributed index).

2.3.2 Το μοντέλο Buzztrack

Μια πρώιμη προσπάθεια για τη δημιουργία ενός συστήματος προσωρινής αποθήκευσης το οποίο χρησιμοποιεί κοινωνικά δεδομένα των χρηστών είναι το Buzztraq και περιγράφεται στο [3].

Μέχρι τώρα τα ΔΠΠ τοποθετούν αντίγραφα των αντικειμένων στους υποκατάστατους εξυπηρετητές με βάση την τοποθεσία των χρηστών. Σύμφωνα με αυτή την πολιτική η γεωγραφική τοποθεσία των προηγούμενων χρηστών οι οποίοι έχουν ζητήσει τα αντικείμενα χρησιμοποιείται για να αποφασιστεί το που θα τοποθετηθούν τα αντίγραφα. Η προτεινόμενη προσέγγιση είναι να χρησιμοποιηθεί η κοινωνική διάδοση. Δηλαδή τα αντίγραφα να τοποθετούνται στις περιοχές που η κοινωνική επιδημία είναι πυκνότερη, σύμφωνα με τις προσβάσεις των φίλων του χρήστη.

Το πρόβλημα αντιμετωπίζεται μέσω Buzztraq είναι:

Έχουμε ένα σύνολο από περιοχές E , ένα σύνολο αντικειμένων O , ένα σύνολο χρηστών U , το γράφο φιλίας του χρήστη G και r τα αιτήματα των χρηστών. Θεωρώντας ότι μπορούμε να έχουμε ένα σταθερό αριθμό K αντιγράφων για κάθε αντικείμενο το Buzztraq δίνει τις K περιοχές στις οποίες πρέπει να τοποθετήσουμε τα αντίγραφα των αντικειμένων έτσι ώστε να ελαχιστοποιούμε τις απομακρυσμένες προσβάσεις.

Η διαίσθηση των συγγραφέων είναι ότι η τοποθέτηση με βάση την θέση που χρησιμοποιεί μόνο τις προσβάσεις των χρηστών αντιδρά αργά όταν οι αιτήσεις στραφούν σε μια νέα περιοχή, επειδή ένας μεγάλος αριθμός των αιτήσεων θα πρέπει να φτάσει από την νέα περιοχή και να εκτοπίσει ένα από τα παλαιότερα αντίγραφα. Αντίθετα, η στρατηγική

πρόβλεψης της κοινωνικής διάδοσης αρχίζει να μετράει τους φίλους των προηγούμενων χρηστών που βρίσκονται στη νέα περιοχή πριν γίνουν αιτήσεις από την νέα περιοχή και θα αντιδράσει ταχύτερα σε μετατόπιση της δημοτικότητας ενός αντικειμένου από μια περιοχή σε άλλη.

Χρησιμοποιώντας το Facebook API ανακτώνται οι φίλοι των χρηστών και οι δημόσιες τους πληροφορίες. Χρησιμοποιούνται 20.740 προφίλ στο Facebook από το δίκτυο του Χάρβαρντ. 21 εκατομμύρια συνδέσεις μεταξύ χρηστών με μέσο βαθμό 63 και μέγιστο βαθμό 911. 1.660 τοποθεσιών εκ των οποίων 1.181 θα αντιστοιχίστηκαν σε γεωγραφικές περιοχές. Για να χωρίσουν τους χρήστες σε περιοχές το Buzztraq χρησιμοποιεί τον αλγόριθμο ομαδοποίησης K-means. Οι τοποθεσίες των χρηστών μετατρέπονται σε γεωγραφικά σημεία μήκους και πλάτους μέσω της υπηρεσίας γεωκωδικοποίησης (geocoding) της Google. Ακολούθως οι χρήστες χωρίζονται σε K ομάδες χρησιμοποιώντας τον K-means όπου η κάθε ομάδα αντιστοιχεί σε μια περιοχή.

Ο τρόπος που επιλύει το πρόβλημα το σύστημα αυτό είναι η χρησιμοποίηση ενός βάρους για κάθε αντικείμενο για κάθε περιοχή και να προτείνει για κάθε αντικείμενο τις K περιοχές στις οποίες έχει το μεγαλύτερο βάρος.

Θεωρώντας ότι έχω U ένα σύνολο από χρήστες U ένα σύνολο από περιοχής R , τον κοινωνικό γράφο G , ένα σύνολο από αντικείμενα O και τις αιτήσεις των χρηστών r ο αλγόριθμος για την επιλογή των αντικειμένων που θα επιλεγθούν είναι ο εξής:

Κατασκευάζουμε ένα βάρος W_{ij} το οποίο αναπαριστά το βάρος του αντικειμένου i στην περιοχή j .

Για οποιοδήποτε αντικείμενο οι K περιοχές με το μεγαλύτερο W_{oj} είναι οι περιοχές στις οποίες το αντικείμενο o θα έχει τις περισσότερες αιτήσεις στο μέλλον.

Αλγόριθμος:

Για κάθε χρήστη u ο οποίος αιτείται το αντικείμενο o

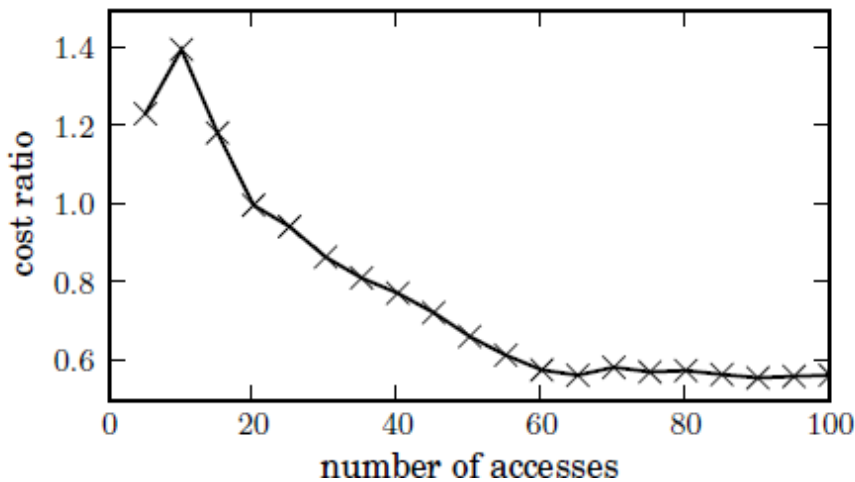
 Πρόσθεσε στην μέχρι τώρα τιμή του W_{oj} για κάθε περιοχή τον αριθμό των φίλων του χρήστη που ανήκουν στην περιοχή j

Επανεξέτασε ποιες περιοχές έχουν το ψηλότερο βάρος W_{oj} και πρότεινε τις K περιοχές με το ψηλότερο βάρος για το αντικείμενο o .

Για την εξέταση της απόδοσης του συστήματος δημιουργείται ένας τεχνητός φόρτος εργασίας όπου ο χρήστης ζητά ένα αντικείμενο λόγω κοινωνικής διάδοσης με πιθανότητα P_s και $1-P_s$ ένα τυχαίο αντικείμενο. Μία απομακρυσμένη πρόσβαση (cache miss) επιλέχθηκε

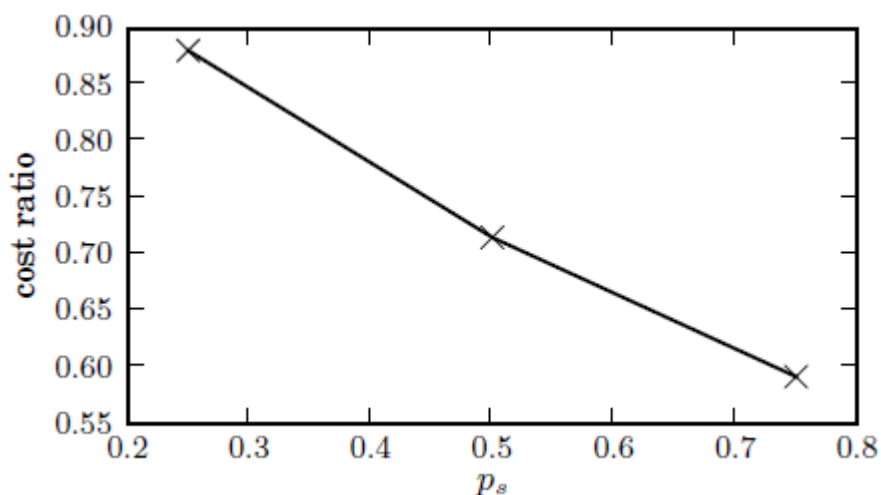
να κοστίζει 20 φορές περισσότερο από την τοπική πρόσβαση(cache hit). Για κάθε αντικείμενο επιτρέπονται 3 αντίγραφα και υπάρχουν 10 περιοχές.

Η απόδοση του συστήματος συγκρίνεται με την τοποθέτηση με βάση τη θέση. Η τοποθέτηση με βάση τη θέση βασίζεται στις θέσεις των προηγούμενων χρηστών για να τοποθετήσει αντίγραφα. Χρησιμοποιώντας ένα συνθετικό φόρτο εργασίας, το σύστημα έχει δοκιμαστεί και έχει βρεθεί να έχει καλύτερες επιδόσεις από την “απλοϊκή” τοποθέτηση με βάση τη θέση.



Σχήμα 2.5 Σύγκριση κόστους μεταξύ τοποθέτησης με βάση τη θέση και κοινωνικής τοποθέτησης

Για $P_s = 0.5$ [3]



Σχήμα 2.6 Μέσο κόστος για διάφορες τιμές του P_s [3]

Πρέπει να σημειωθεί ότι η παρακολούθηση των προσβάσεων των χρηστών και η αντιστοίχιση αυτών με τα αντίστοιχα προφίλ δεν φαίνεται εφικτή προσέγγιση. Το σύστημα προτείνει να χρησιμοποιείται μια κεντρική σελίδα όπου οι χρήστες να συνδεονται για να δουν βίντεο και να έχει πρόσβαση στο κοινωνικό προφίλ τους. Αυτά τα 2 είναι δύσκολο να

συμπέσουν, για παράδειγμα οι περισσότεροι χρήστες χρησιμοποιούν Facebook για κοινωνική δικτύωση και το YouTube της Google για την προβολή βίντεο. Επίσης, πολλοί χρήστες θα κρίνουν τη σύνδεση σε έναν κεντρικό χώρο για δουν περιεχόμενο πολυμέσων περιττή. Επίσης, η έννοια της κοινωνικής διάδοσης είναι έμμεση και δεν παρακολουθούνται ή αναλύονται κοινωνικές διαδόσεις στο Buzztraq.

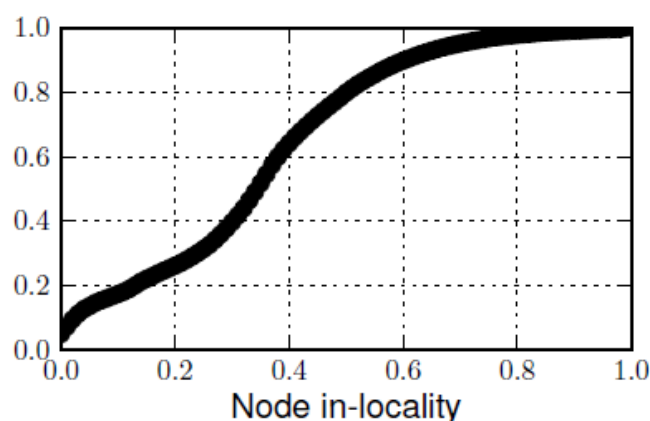
2.3.3 Τοποθέτηση με βάση την απόσταση μεταξύ των χρηστών

Το πρώτο ολοκληρωμένο σύστημα το οποίο χρησιμοποιεί κοινωνικές διαδόσεις για αποφασίζει που θα αντιγραφεί περιεχόμενο περιγράφεται στο[4]. Το γενικό πρόβλημα στο οποίο απευθύνεται είναι το πώς να χρησιμοποιηθούν οι κοινωνικές διαδόσεις και οι πληροφορίες από τα κοινωνικά δίκτυα για να βελτιωθεί η προσωρινή αποθήκευση από τα ΔΠΠ.

Για να αποφασιστεί εάν είναι δυνατό να υπολογίσουμε κατά πόσο οι κοινωνικές διαδόσεις εξαπλώνονται παγκόσμια ή τοπικά από τις γεωκοινωνικές (Geosocial) ιδιότητες των χρηστών οι οποίοι λαμβάνουν μέρος στην κοινωνική διάδοση έχει δημιουργηθεί η μετρική τοπικότητα (locality) ενός κόμβου. Η τοπικότητα ενός κόμβου είναι μία μετρική του πόσο γεωγραφικά κοντά είναι οι ένας χρήστης στους φίλους τους και ορίζεται ως:

$$\frac{1}{k_v} \sum_{j \in \Gamma_v} e^{-l_{vj}/\beta}$$

l_{vj} είναι το βάρος της ακμής μεταξύ των κόμβων v και j (η απόσταση σε km) και το β είναι ένας συντελεστής κλίμακας Το β θέτεται έτσι ώστε όταν 2 κόμβοι είναι 1000km κοντά η τοπικότητα του κόμβου να είναι ίση με 0.6.



Σχήμα 2.7 Αθροιστική συνάρτηση κατανομής της τοπικότητας των κόμβων [4]

Αυτό το γράφημα δίνει την εντύπωση ότι λίγοι χρήστες έχουν οπαδούς με σε τοποθεσία κοντά στη δική τους, αλλά αποδεικνύεται ότι οι χρήστες στην πραγματικότητα διαδίδουν περιεχόμενο στους οπαδούς που είναι κοντά στη θέση τους.

Το δίκτυο του twitter είχε ιχνηλατηθεί και είχαν αποκτηθεί τα tweets, η τοποθεσία και η λίστα ακολούθων για 409,093 χρήστες. Είχαν συλλεχθεί 334 εκατομμύρια μηνύματα εκ των οποίων 3 εκατομμύρια περιείχαν σύνδεσμο στο YouTube με 1,111,586 διαφορετικά βίντεο.

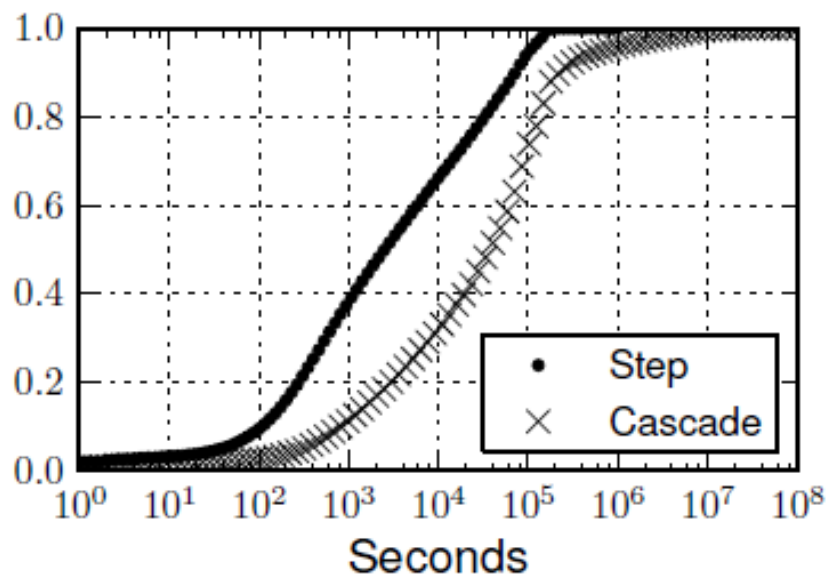
Οι κοινωνικές διαδόσεις αναπαρίστανται με ένα δέντρο όπου ακμή από το χρήστη A στο χρήστη B σημαίνει ότι ο χρήστης B έχει λάβει περιεχόμενο από τον χρήστη A. Σύμφωνα με τις τοποθεσίες των χρηστών οι ακμές έχουν ως βάρος την απόσταση μεταξύ των χρηστών.

Χρησιμοποιώντας το δέντρο διάδοσης υπολογίζονται 2 μετρικές:

Το geodiversity το οποίο είναι ο μέσος όρος των αποστάσεων μεταξύ όλων των χρηστών στο δέντρο και το georange το οποίο είναι ο μέσος όρος των αποστάσεων μεταξύ όλων των χρηστών και του χρήστη ρίζα του δέντρου.

Χαρακτηριστικά των κοινωνικών διαδόσεων:

Από τις 84,337 συνολικές διαδόσεις 60,000 ήταν μεταξύ μόνο 2 κόμβων αλλά κάποιες είχαν φτάσει πάνω από 4000 κόμβους. Οι μικρότερες διαδόσεις δεν είναι λιγότερο σημαντικές εφόσον ένας χρήστης μπορεί να έχει μεγάλο ακροατήριο το οποίο δεν μεταβιβάζει απαραίτητα το περιεχόμενο αλλά μπορεί να το βλέπει.



Σχήμα 2.8 Συνάρτηση αθροιστικής κατανομής του χρόνου καθυστέρησης μεταξύ 2 συνεχόμενων tweets και τη συνολική διάρκεια διάδοσης [4]

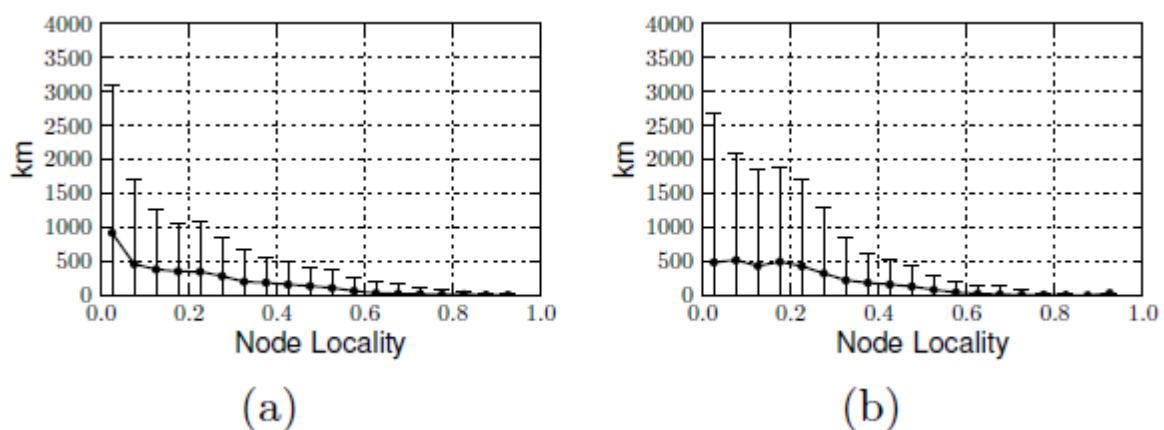
40% των διαδόσεων έχουν μια καθυστέρηση 15 λεπτών και 10% καθυστέρηση των 2 λεπτών.

Επίσης, το 80% των διαδόσεων τελειώνουν εντός 24 ωρών με 30% των διαδόσεων λήγει σε λιγότερο από 3 ώρες. Αυτό δείχνει ότι οι διαδόσεις μπορεί να εξαπλωθούν πολύ γρήγορα κάποια στιγμή στη ζωή τους, αν και μερικά βήματα για να κάνουμε ώρες για να συμβεί. Η ταχεία εξάπλωση και η σύντομη διάρκεια ζωής των διαδόσεων πρέπει να ληφθούν υπόψη κατά τη δημιουργία ενός συστήματος που προσπαθεί να τις εκμεταλλευτούν. Επίσης, αν και μια διάδοση θα μπορούσε να τερματίσει σε κάτω από 24 ώρες τα αποτελέσματά της μπορεί να χρειαστούν μέρες για να εξαλειφθούν λαμβάνοντας υπόψη τους χρήστες που εκτίθενται στο περιεχόμενο, αλλά δεν είναι μέρος της διάδοσης και έτσι μπορούν να δουν το περιεχόμενο σε μεταγενέστερο χρόνο.

Τα βήματα που γίνονται σε μια διάδοση γίνονται σε σχετικά μικρές αποστάσεις. 60% του των βημάτων γίνονται σε απόσταση 5.000 χιλιομέτρων το 30% είναι λιγότερο από 1.000 χιλιόμετρα και το 20% γίνονται λιγότερο από 100 χιλιόμετρα. Διαπιστώθηκε ότι το 40% των geodiversity και 90% των τιμών georange είναι λιγότερο από 1000 χιλιόμετρα.

Αυτό δείχνει ότι τα περισσότερα tweets έχουν την τάση να εξαπλώνονται σε μικρές αποστάσεις και να παραμείνουν κοντά στον αρχικό αποστολέα.

Τέλος, παρουσιάστηκε σχέση μεταξύ της τοπικότητας των πρώτων χρηστών και το GeoRange της διάδοσης, που απεικονίζεται παρακάτω. (α: για τον πρώτο κόμβο, β: για τους πρώτους 2 κόμβους)



Σχήμα 2.9 Georange [4]

Υπάρχει σαφής συσχέτιση μεταξύ της τοπικότητας και του GeoRange δείχνοντας ότι όταν ο αρχικός χρήστης της διάδοσης έχει μεγαλύτερη τοπικότητα η διάδοση εξαπλώνεται σε

μικρότερη περιοχή. Έτσι μπορούμε να συμπεράνουμε αν μια διάδοση θα εξαπλωθεί σε τοπικό ή παγκόσμιο επίπεδο

Περιγραφή συστήματος:

Η ιδέα που χρησιμοποιείται είναι ότι όταν το περιεχόμενο μεταδίδεται από χρήστες με υψηλή τοπικότητα τότε είναι πιο πιθανό να εξαπλωθεί τοπικά σε μια περιοχή και έτσι μελλοντικές αιτήσεις θα φτάσουν από την ίδια περιοχή. Για να χρησιμοποιηθεί αυτή η πληροφορία δημιουργούνται 2 βάρη για το περιεχόμενο, το Geosocial όπου αθροίζεται η τοπικότητα όλων των κόμβων που μοιράστηκαν το συγκεκριμένο αντικείμενο (ακόμα και αν αυτοί δεν εμπλέκονταν σε κοινωνική διάδοση) και το Geocascade όπου αθροίζεται η τοπικότητα όλων των κόμβων που συμμετέχουν στην κοινωνική διάδοση για το συγκεκριμένο αντικείμενο. Η προτεραιότητα του βίντεο πολλαπλασιάζεται με ένα από αυτά τα βάρη για να το εμπλουτίσει με κοινωνική πληροφορία. Το περιεχόμενο με υψηλή τοπικότητα παίρνει ψηλό σκορ και παραμένει στον υποκατάστατο εξυπηρετητή.

Ένα ΔΠΠ είχε μοντελοποιηθεί και δοκιμαστεί. Για τη προσομοίωση είχαν χρησιμοποιηθεί οι ιδιότητες του ΔΠΠ Limelight με 19 συμπλέγματα εξυπηρετητών σε όλο τον κόσμο και η χωρητικότητα του κάθε εξυπηρετητή είναι ίση με το 0.01% του συνολικού καταλόγου. Για να δοκιμαστεί το σύστημα υιοθετούνται 3 προσεγγίσεις προσωρινής αποθήκευσης, η πιο πρόσφατα χρησιμοποιημένο (Last Recently Used), η πιο λίγο συχνά χρησιμοποιημένο (Least Frequently Used) και μία υβριδική προσέγγιση που συνδυάζει και τα 2. Αυτές οι μετρικές συνδυάζονται με τα βάρη που προαναφέρθηκαν.

Το σύστημα δείχνεται ότι έχει καλύτερη απόδοση όταν δεν χρησιμοποιείται κάποιο βάρος. Καλύτερη βελτίωση δίνεται με το Geocascade βάρος.

Αποτελέσματα:

Συγκρίνεται η απόδοση των διάφορων στρατηγικών σε σχέση με τη μη χρησιμοποίηση κάποιου βάρους. (cache hits)

- Geosocial-LRU 55% αύξηση
- Geocascade-LRU 70% αύξηση
- Geosocial-LFU 50% αύξηση
- Geocascade-LFU 70% αύξηση
- Geosocial-MIXED 50% αύξηση
- Geocascade-MIXED 65% αύξηση

Η αύξηση της απόδοσης είναι μικρότερα όταν το μέγεθος της χωρητικότητας των υποκατάστατων εξυπηρετητών αυξάνεται.

Το μοντέλο έχει βελτιώσει τις παραδοσιακές τεχνικές προσωρινής αποθήκευσης για περιεχόμενο βίντεο, αλλά χωρίς να εισάγει νέες τεχνικές για προανάκληση(prefetching) ή την ανακάλυψη τοπικά δημοφιλή αντικειμένων πριν γίνουν δημοφιλή. Η κοινωνική διάδοση συμβάλλει σε ορισμένα αντικείμενα να παραμένουν στους κεντρικούς υπολογιστές χωρίς να αφαιρείται λανθασμένα υπέρ κάποιου άλλου νεότερα δημοφιλή αντικείμενου. Επίσης, τα παγκοσμίως δημοφιλή αντικείμενα δεν διαφοροποιείται από τα αντικείμενα που έχουν μόνο τοπική δημοτικότητα και αντιμετωπίζονται με το ν ίδιο τρόπο. Επίσης, το σύστημα δεν προβλέπει εξισορρόπηση φορτίου μεταξύ περιφερειακών συμπλέγματα. Μικρότερα συμπλέγματα καταλήγουν να εξυπηρετούν διπλάσιες αιτήσεις από συμπλέγματα που έχουν 3 ή 4 φορές το μέγεθός τους.

Και πάλι ο συνθετικός φόρτος εργασίας που χρησιμοποιούνται για την αξιολόγηση του συστήματος μπορεί να έχει διαφορετικά αποτελέσματα από τα πραγματικά αποτελέσματα που θα μπορούσε να παρατηρηθεί σε ένα πραγματικό CDN με πραγματικά αιτήματα

2.3.4 Αντιμετώπιση του long tail περιεχομένου

Στο [11] περιγράφεται ένα καινοτόμο σύστημα το οποίο χρησιμοποιεί πληροφορίες για τη συμπεριφορά των χρηστών και τις αξιοποιεί προς όφελος του ΔΠΠ.

Χρησιμοποιώντας ένα μεγάλο σύνολο από τουιτς είχε γίνει ανάλυση και επιβεβαιώθηκε η long tail κατανομή της διάδοσης. 30% του περιεχομένου θεάθηκε λιγότερο από 500 φορές. Αυτό επικυρώνει ότι υπάρχει η ανάγκη να χειριστεί το περιεχόμενο που ανήκει στο long tail.

Οι τοποθεσίες των χρηστών ομαδοποιούνται και υπολογίζεται ότι ένας χρήστης έχει κατά μέσο όρο 19.72 ακολούθους στη δική του περιοχή και 8.91 ακολούθους σε διαφορετικές περιοχές δείχνοντας ότι περιεχόμενο που μοιράζεται από ένα χρήστη είναι πιθανό να ζητηθεί και σε άλλες περιοχές.

Μία ανάλυση των αιτήσεων Http έδωσε την πιθανότητα ότι 9.8 % των συνδέσμων που μοιράζονται ζητούνται από τους χρήστες. Αυτό δίνει την πιθανότητα με την οποία ένας φίλος κάποιου χρήστη θα ζητήσει κάποιο αντικείμενο το οποίο έχει μοιραστεί ο φίλος του. Ακολούθως χρησιμοποιώντας το γεγονός ότι οι αιτήσεις των χρηστών ακολουθούν ημερήσιες τάσεις περιγράφεται για το πώς να δημιουργηθούν τεχνητά δεδομένα αιτήσεων ώστε να γίνεται γνωστό πότε θα ζητηθεί το περιεχόμενο.

Το tailgate είναι διαφορετικό από τα άλλα συστήματα ως προς το σκοπό. Το tailgate είναι ένα σύστημα προγραμματισμού για το πότε να μεταφέρεται το περιεχόμενο στις διάφορες τοποθεσίες στον παγκόσμιο ιστό κατά τις ώρες που δε υπάρχει συμφόρηση στο δίκτυο. Το tailgate χρησιμοποιεί το γεγονός ότι οι χρήστες ακολουθούν ημερήσια μοτίβα όταν πλοηγούνται στον παγκόσμιο ιστό. Η δραστηριότητα των χρηστών αυξάνεται συγκεκριμένες ώρες κατά τη διάρκεια της μέρας και περιορίζεται τη νύχτα. Αυτές οι ώρες είναι διαφορετικές για διαφορετικές περιοχές στον κόσμο. Εφόσον το σύστημα αποφεύγει να μεταφέρει δεδομένα στις ώρες αιχμής το κόστος είναι μειωμένο εφόσον το κόστος σε αυτές τις ώρες μειώνεται. Επίσης επειδή λαμβάνει υπόψη τα μοτίβα γραψίματος και διαβάσματος των χρηστών στο παγκόσμιο ιστό το σύστημα απο δδει καλά χωρίς επαρκή κοινωνική πληροφορία.

Λειτουργία:

Υποθέτοντας ότι ένας χρήστης ανεβάζει περιεχόμενο. Λαμβάνοντας υπόψη τις τοποθεσίες των φίλων του χρήστη και τις κατά προσέγγιση ώρες που γίνονται τα διαβάσματα στην κάθε τοποθεσία βρίσκει τη συντομότερη ώρα που είναι πιθανό να ζητηθεί το περιεχόμενο σε κάθε τοποθεσία. Το επόμενο βήμα είναι να προγραμματιστεί η μεταφορά του περιεχομένου σε κάποια ώρα στην οποία υπάρχει η ελάχιστη συμφόρηση. Για αυτό ο χρόνος χωρίζεται σε καλάθια των 5 λεπτών και εντοπίζεται το καλάθι που έχει τις πιο λίγες μεταφορές από τη στιγμή που γίνεται το ανέβασμα μέχρι στη στιγμή που υπολογίζεται ότι θα αιτηθεί για πρώτη φορά το περιεχόμενο.

Ακόμα μία καινοτομία που παρουσιάζεται είναι στην αξιολόγηση του συστήματος. Για να αξιολογηθεί το σύστημα οι αστοχίες της προσωρινής μνήμης σε σχέση με το κόστος των μεταφορών μεταξύ των κέντρων δεδομένων λαμβάνεται υπόψη το οποίο δίνει καλύτερη άποψη για την αποδοτικότητα του συστήματος από την απλοϊκή προσέγγιση του να μετρούνται οι αστοχίες της προσωρινής μνήμης. Το σύστημα αυτό αποδίδει πάρα πολύ καλά σε κόστος και μειώνει τις αστοχίες της προσωρινής μνήμης.

Κεφάλαιο 3

Συλλογή Δεδομένων

3.1 Συλλογή κοινωνικών δεδομένων	19
3.2 Εξαγωγή των πραγματικών συνδέσμων από τα Tweets	22
3.3 Εύρεση της δημοτικότητας των βίντεο στο YouTube	23
3.4 Εύρεση της τοποθεσίας των χρηστών	24
3.5 Προβλήματα κατά τη συλλογή δεδομένων	24
3.6 Σχηματική αναπαράσταση της Ιχνηλασίας των δεδομένων	25

4.1 Συλλογή Κοινωνικών Δεδομένων

Βασική προϋπόθεση στη υλοποίηση ενός συστήματος που εξάγει πληροφορίες για τους χρήστες είναι η ύπαρξη κοινωνικών δεδομένων.

Έχουμε ιχνηλατήσει τα κοινωνικά δίκτυα Twitter για να συλλέξουμε τα κοινωνικά δεδομένα. Έχουμε επιλέξει το κοινωνικό δίκτυο Twitter διότι λόγω της φύσης του τα δεδομένα είναι πολύ πιο ελεύθερα. Το Twitter έχει καθιερωθεί σαν ένα κοινωνικό μέσο μαζικής ενημέρωσης όπου οι προσωπικότητες ενημερώνουν για τις πράξεις τους, τις απόψεις τους και επικοινωνούν με το υπόλοιπο δίκτυο. Κατά βάση τα δεδομένα στο Twitter είναι ελεύθερα εκτός και αν ο χρήστης επιλέξει ρητά να τα αποκρύψει αντίθετα με άλλα κοινωνικά δίκτυα όπου τα δεδομένα των χρηστών μπορούν να αποκτηθούν μόνο μετά από έγκριση του ιδίου του χρήστη ή του κοινωνικού δικτύου.

Για την ανάκτηση των δεδομένων έχουμε χρησιμοποιήσει το API που προσφέρει το Twitter για αυτό το σκοπό [18]. Μέσω του API μία εφαρμογή μπορεί να έχει πρόσβαση στα tweets του χρήστη και σε άλλες πληροφορίες και ακόμα να δημοσιεύσει tweets για αυτόν. Εμείς χρησιμοποιούμε το API για να εξάγουμε δημόσιες πληροφορίες από χρήστες οι οποίοι δεν έχουν δώσει οποιαδήποτε δικαιώματα στην εφαρμογή μας.

Για κάθε χρήστη παίρνουμε τριών ειδών πληροφορίες. Τον κοινωνικό του γράφο, δηλαδή τη λίστα με τους ακολούθους του, διάφορες πληροφορίες για αυτόν όπως τοποθεσία, αριθμός δημοσιεύσεων κτλ και τις δημοσιεύσεις του, δηλαδή τα tweets που έχει κάνει.

Το Twitter επιβάλλει τον περιορισμό των 150 αιτήσεων για μη εξουσιοδοτημένες εφαρμογές και 350 αιτήσεις για τις εγκεκριμένες εφαρμογές, ανά ώρα (εκτός αν το Twitter δώσει επιπλέον δικαιώματα). Μία αίτηση μπορούν να προσκομίσει μέχρι και 5000 ακόλουθους για ένα χρήστη, ή 200 tweets για ένα χρήστη, ή τις πληροφορίες για 200 χρήστες.

Ζητήσαμε από τους φίλους και συναδέλφους να εξουσιοδοτήσουν την εφαρμογή μας ώστε να αυξηθούν οι επιτρεπόμενες αιτήσεις μας χωρίς να παραβιάζονται οι κανόνες του Twitter API.

Τρόπος συλλογής:

Για να συλλέξουμε τα δεδομένα χρησιμοποιήσαμε τη βιβλιοθήκη για το twitter από το [19]. Η βιβλιοθήκη υλοποιεί κάποιες βασικές λειτουργίες του API και είναι υλοποιημένη σε γλώσσα προγραμματισμού Python. Χρησιμοποιώντας τη βιβλιοθήκη αυτή δημιουργήσαμε τον δικό μας ιχνηλάτη. Τρέξαμε τον ιχνηλάτη για 5 μήνες και συλλέξαμε τα πλείστα δεδομένα των χρηστών.

Τους χρήστες για τους οποίους συλλέξαμε τα δεδομένα τους πήραμε από μία ιχνηλασία από το 2009 από το [20]. Η ιχνηλασία αυτή περιελάμβανε ολόκληρο το κοινωνικό δίκτυο του Twitter μέχρι το 2009. Χρησιμοποιήσαμε αυτούς τους χρήστες έτσι ώστε να ζητήσουμε χρήστες που σίγουρα είναι στο κοινωνικό δίκτυο του Twitter διότι σε διαφορετική περίπτωση θα έπρεπε να ζητούμε χρήστες επιλέγοντας τυχαίες ταυτότητες με αποτέλεσμα ο χρόνος ανάκτησης των δεδομένων να αυξάνεται κατά πολύ.

Τρέξαμε τον ιχνηλάτη σε διάφορους εξυπηρετητές και τοποθεσίες. Χρησιμοποιήσαμε την υποδομή του πανεπιστημίου, δωρεάν υποδομές από εταιρείες παροχής cloud και σε δικούς μας υπολογιστές. Χρησιμοποιήσαμε 3 εξυπηρετητές στο πανεπιστήμιο Κύπρου, την υπηρεσία cloud computing Ωκεανός, την υπηρεσία Amazon EC2 Cloud, το Rackspace Datacenters και 4 ιδιόκτητους υπολογιστές.

Περιγραφή δεδομένων:

Ο κοινωνικός γράφος αποτελείται από 37,343,273 χρήστες 6,820,494,777 ακμές.

Οι πληροφορίες που έχουμε συλλέξει για κάθε χρήστη είναι οι εξής:

Id	Ταυτότητα του χρήστη
Verified	Αν ο χρήστης έχει επιβεβαιώσει την ηλεκτρονική του διεύθυνση
followers_count	Αριθμός ακολούθων του χρήστη
Protected	Εάν οι πληροφορίες του χρήστη είναι προσωπικές
listed_count	Σε πόσα tweets αναφέρεται ο χρήστης
statuses_count	Πόσα tweets έχει δημοσιεύσει ο χρήστης
friends_count	Πόσους χρήστες ακολουθεί ο χρήστης
Location	Δηλωμένη τοποθεσία του χρήστη
geo_enabled	Εάν είναι ενεργοποιημένη η υπηρεσία που δίνει την τοποθεσία του χρήστη μαζί με κάθε tweet
Lang	Γλώσσα του χρήστη
favourites_count	Πόσα tweets ο χρήστης έχει προσθέσει στα αγαπημένα του
created_at	Πότε δημιουργήθηκε ο λογαριασμός του χρήστη
time_zone	Ζώνη ώρας στην οποία βρίσκεται ο χρήστης
time_zone	Διαφορά ώρας μεταξύ της ζώνης που βρίσκεται ο χρήστης και της ζώνης UTC σε δευτερόλεπτα

Τα tweets που έχουμε συλλέξει για τους χρήστες αντιστοιχούν στο 70% των χρηστών. Για τους 1,384,758 χρήστες για τους οποίους τρέξαμε τα πειράματά μας έχουμε συλλέξει 299,071,571 tweets.

Οι πληροφορίες που συλλέξαμε για κάθε tweet είναι οι εξής:

Id	Ταυτότητα του Tweet
Text	Κείμενο του tweet
created_at	Πότε δημοσιεύτηκε το tweet
Retweeted	Εάν είναι επαναδημοσίευση
in_reply_to_status_id	Η ταυτότητα του tweet στο οποίο είναι απάντηση
in_reply_to_user_id	Η ταυτότητα του χρήστη στον οποίο απαντά
urls	Οι συνδέσμοι του tweet
retweet_count	Πόσες φορές έχει γίνει επαναδημοσίευση

3.2 Εξαγωγή των πραγματικών συνδέσμων από τα Tweets

Λόγω του υποχρεωτικού ορίου χαρακτήρων το οποίο επιβάλλει το Twitter για κάθε δημοσίευση οι σύνδεσμοι στα tweets είναι σμικρυνσμένοι. Για αυτή τη λειτουργία χρησιμοποιούνται υπηρεσίες σμίκρυνσης συνδέσμων όπως το bit.ly και το owl.ly. Αυτό δημιουργεί την ανάγκη να εξαχθούν οι πραγματικοί σύνδεσμοι για να μπορέσουμε να εντοπίσουμε το περιεχόμενο στο οποίο δείχνουν.

Ο πιο απλός τρόπος για να γίνει αυτό είναι ανοίγοντας τους συνδέσμους ακολουθώντας τυχόν ανακατευθύνσεις και να καταλήξουμε στην πραγματική διεύθυνση στην οποία δείχνει ο σύνδεσμος. Αυτή η προσέγγιση είναι χρονοβόρα και δεν μπορεί να χειριστεί νεκρές διεθύνσεις. Μία βελτιωμένη προσέγγιση είναι να ζητούμε την επικεφαλίδα της διεύθυνσης και να ελέγχουμε από τους κωδικούς που επιστρέφει το http αν υπάρχει ανακατεύθυνση και μόνο τότε να ζητούμε ολόκληρη τη σελίδα γλυτώνοντας χρόνο και bandwidth. Αυτή η προσέγγιση έχει το μειονέκτημα ότι πολλές φορές παρόλο που έχουμε φτάσει στην τελική σελίδα στη οποία δείχνει ο σύνδεσμος ο http κωδικός δείχνει ανακατεύθυνση λανθασμένα

και οδηγούμαστε σε μια τελειώς διαφορετική σελίδα ή ακόμα η σελίδα έχει ανακατεύθυνση στον εαυτό της και δημιουργείται ένας άπειρος βρόγχος.

Η καλύτερη λύση για την εύρεση των πραγματικών συνδέσμων είναι η χρησιμοποίηση ειδικών υπηρεσιών που μπορούν να επεκτείνουν ένα σμικρυνσμένο σύνδεσμο.

Σε αυτές τις υπηρεσίες γίνεται μία αίτηση http με τον σύνδεσμο που θέλουμε να επεκτείνουμε και δίνεται ως απάντηση το πραγματικό URL.

Για παράδειγμα εάν κάνουμε αυτή την αίτηση στην υπηρεσία untiny.me

`http://untiny.me/api/1.0/extract?url=http://bit.ly/z8bMmH` θα μας δοθεί ως απάντηση ο πραγματικός σύνδεσμος σε μορφή XML

```
<response>
```

```
<org_url>http://www.youtube.com/watch?v=TBHucXOBlrU</org_url>
```

```
</response>
```

Έχουμε δημιουργήσει ένα ιχνηλάτη σε γλώσσα προγραμματισμού Python ο οποίος χρησιμοποιεί τα API των υπηρεσιών στο [21] και [22] για να εξάγει τις πραγματικές διευθύνσεις των βίντεο.

Λόγω του ότι πολλοί σύνδεσμοι στο twitter είναι 2 φορές σμικρισμένοι πολλές υπηρεσίες δεν μπορούν να εξάγουν την πραγματική διεύθυνση και αυτές που το κάνουν απαντούν πολύ αργά στα αιτήματα μας έχουμε καταφέρει να αναλύσουμε τις δημοσιεύσεις μόνο για 1,384,758 χρήστες για τους οποίους και θα τρέξουμε τα πειράματα μας.

3.3 Εύρεση της δημοτικότητας των βίντεο στο YouTube

Βασική προϋπόθεση για την δημιουργία ενός συστήματος που βασίζεται πάνω σε πραγματικά δεδομένα είναι να γνωρίζουμε πόσο δημοφιλή είναι τα βίντεο στον πραγματικό κόσμο. Η δημοτικότητα των βίντεο στο YouTube μπορεί να εξαχθεί από το πόσες θεάσεις έχει ένα βίντεο, σε πόσους χρήστες τους αρέσει και σε πόσους δεν αρέσει και επίσης από τον αριθμό των σχολίων στο βίντεο. Χρειάζεται να ξέρουμε εάν η πραγματική δημοτικότητα σχετίζεται με την δημοτικότητα στο κοινωνικό δίκτυο.

Για να εξάγουμε τις θεάσεις ενός βίντεο χρησιμοποιούμε το YouTube API [23]. Μέσω του API και της υπηρεσίας Google Analytics μπορούμε να ζητήσουμε διάφορα στατιστικά για οποιοδήποτε βίντεο στο YouTube το οποίο είναι ακόμα ενεργό, δηλαδή το βίντεο υπάρχει ακόμα στη βάση του YouTube και ο ιδιοκτήτης του βίντεο επιτρέπει τα στατιστικά του βίντεο να είναι διαθέσιμα σε όλους. Για κάθε βίντεο εξάγουμε τον αριθμό θεάσεων και την χρονική του διάρκεια.

Στα 299,071,571 tweets που έχουμε αναλύσει έχουμε βρει ότι έχουν μοιραστεί 2,198,287 σύνδεσμοι με βίντεο στο YouTube. Άρα 0.7% των tweets περιέχουν ένα σύνδεσμο σε βίντεο στο YouTube. Από τους συνδέσμους αυτούς έχουν εξαχθεί 1,422,096 μοναδικά βίντεο και 776,191 έχουν μοιραστεί 2 ή περισσότερες φορές. Αυτό μας δείχνει ότι υπάρχει ένας πολύ μεγάλος αριθμός βημάτων κοινωνικής διάδοσης.

Έχουμε υλοποιήσει μια εφαρμογή σε γλώσσα PHP η οποία ζητά τις θεάσεις και τη χρονική διάρκεια για κάθε βίντεο που έχουμε εξάγει από τα tweets. Λόγω του ότι κάποια βίντεο έχουν αφαιρεθεί από το YouTube ή τα στατιστικά τους δεν είναι διαθέσιμα έχουμε συλλέξει πληροφορίες για 1,113,294

3.4 Εύρεση της τοποθεσίας των χρηστών

Οι χρήστες δηλώνουν την τοποθεσία τους στο Twitter. Για να μπορέσουμε να εξάγουμε πληροφορίες από τις τοποθεσίες των χρηστών πρέπει να αντιστοιχίσουμε τα ονόματα των τοποθεσιών των χρηστών σε γεωγραφικές συντεταγμένες. Για να το κάνουμε αυτό χρησιμοποιούμε το Geocoding API του Google Maps[24] . Μπορούμε να κάνουμε μία http αίτηση στο API και να ζητήσουμε τις γεωγραφικές συντεταγμένες οποιασδήποτε τοποθεσίας. Έχουμε υλοποιήσει μια εφαρμογή σε γλώσσα python η οποία αιτείται τις γεωγραφικές συντεταγμένες για τις τοποθεσίες τις οποίες δηλώνουν οι χρήστες. Λόγω του περιορισμού στον αριθμό αιτήσεων από την Google χρησιμοποιήσαμε proxy servers για να αυξήσουμε τον αριθμό των αιτήσεων που δικαιούμαστε. Η δεύτερη τεχνική που χρησιμοποιήσαμε ήταν ένα λεξικό στο οποίο αποθηκεύουμε όλες τις τοποθεσίες που συναντήσαμε μέχρι τώρα και τις αντιστοιχίζουμε τους συντεταγμένες κάνοντας πολύ λιγότερες αιτήσεις και έτσι βρίσκοντας τις γεωγραφικές συντεταγμένες πολύ περισσότερων χρηστών.

3.5 Προβλήματα κατά τη Συλλογή Δεδομένων

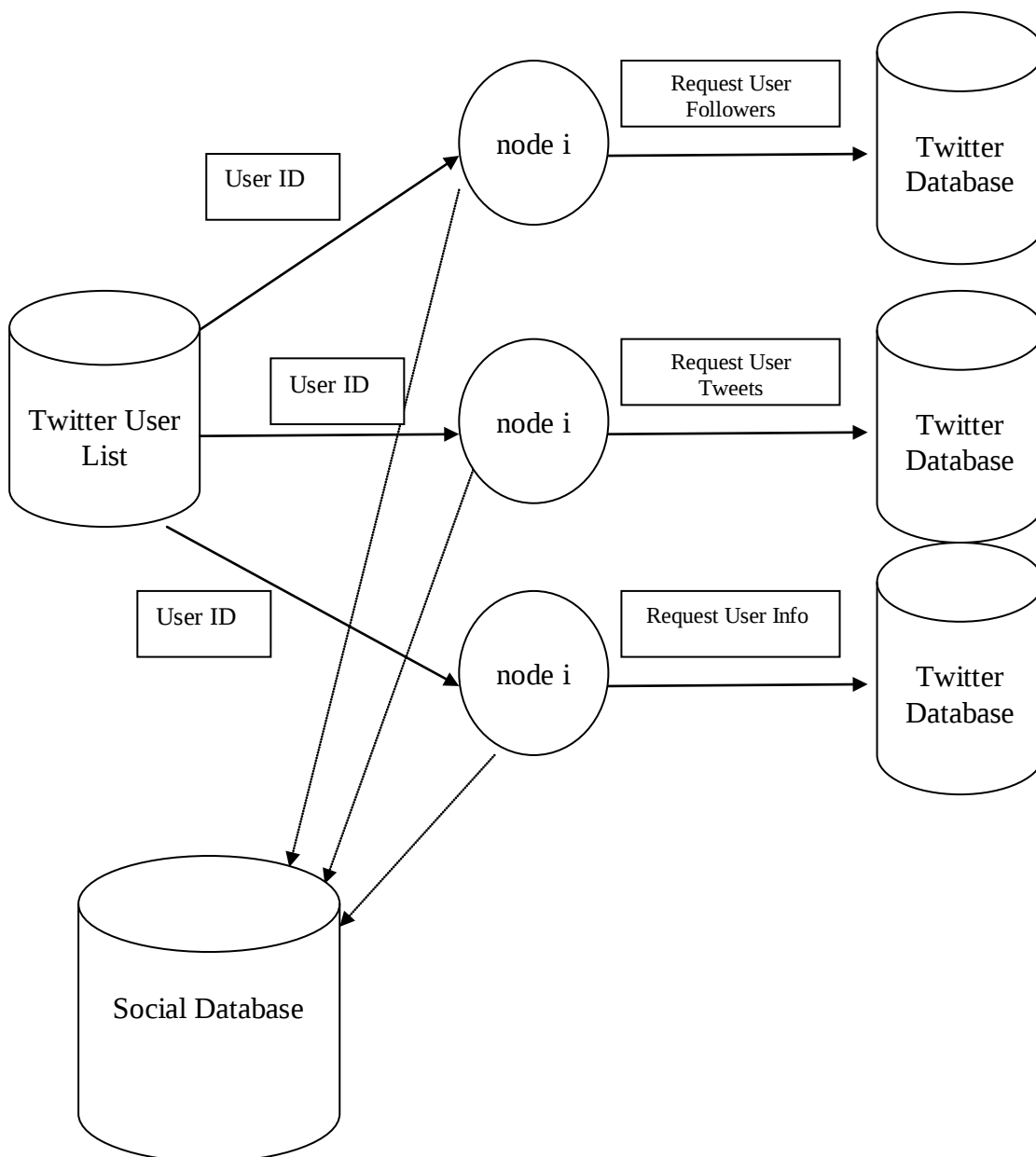
Το κοινωνικό δίκτυο του twitter επιβάλλει περιορισμό στον αριθμό των αιτήσεων που μπορούν να γίνουν ανά ώρα ανά IP. Για κάθε χρήστη όμως που εξουσιοδοτεί μια εφαρμογή επιτρέπονται ακόμα 350 αιτήσεις ανά ώρα. Έτσι ζητήσαμε από φίλους και συναδέλφους να εξουσιοδοτήσουν την εφαρμογή μας αυξάνοντας τον αριθμό των επιτρεπομένων αιτήσεων. Επίσης τρέξαμε τον ιχνηλάτη σε διάφορους εξυπηρετητές διότι το όριο μετρούσε ανά IP.

Για την αντιστοίχιση των περιοχών των χρηστών με γεωγραφικές συντεταγμένες χρησιμοποιήσαμε το Geocoding service του Google Maps []. Αυτή η υπηρεσία επιβάλλει αυστηρό όριο ανά μέρα στον αριθμό των αιτήσεων που μπορούν να γίνουν από κάποιο IP.

Για να προσπεράσουμε το ν περιορισμό αυτό τρέξαμε την εφαρμογή μας από πολλούς εξυπηρετητές μέσω proxies.

Λόγω του ότι πολλοί σύνδεσμοι στο twitter είναι 2 φορές σμικρισμένοι πολλές υπηρεσίες δεν μπορούν να εξάγουν την πραγματική διεύθυνση και αυτές που το κάνουν απαντούν πολύ αργά στα αιτήματα μας δεν είχαμε καταφέρει να αναλύσουμε τις δημοσιεύσεις όλων των χρηστών. Μόνο για 1,384,758 χρήστες για τους οποίους και τρέξαμε τα πειράματά μας.

3.6 Σχηματική Αναπαράσταση της Ιχνηλασίας των Δεδομένων



Σχήμα 3.1 Σύστημα ιχνηλασίας δεδομένων

Κεφάλαιο 4

Ανάλυση των Διαδόσεων

4.1 Συντελεστής διάδοσης	26
4.2 Συντελεστής διάδοσης σχετικά με τις ζώνες ώρας	28
4.3 Συντελεστής διάδοσης σχετικά με τον αριθμό των φίλων που διαδίδουν	29
4.4 Καθυστέρηση διάδοσης	30
4.5 Σχέση μεταξύ κοινωνικής διάδοσης και δημοτικότητας	31

4.1 Συντελεστής Διάδοσης

Για να ελέγξουμε κατά πόσο επηρεάζονται οι χρήστες από τις κοινωνικές διαδόσεις υπολογίσαμε το συντελεστή διάδοσης. Από τους 1,384,758 αρχικούς χρήστες για τους οποίους έχουμε τα tweets τους μέσω κοινωνικής διάδοσης έχουν αποδεκτεί τα μηνύματα τους 12,384,109. 590,310 είναι και αποδέκτες και αποστολείς περιεχομένου.

Έχουμε βρει το συντελεστή διάδοσης για διαφορετικές χρονικές περιόδους και για τις διαφορετικές σχέσεις μεταξύ των χρηστών. Ο συντελεστής διάδοσης δείχνει με ποια πιθανότητα κάποιος χρήστης που βλέπει περιεχόμενο μέσα από το κοινωνικό δίκτυο θα το μοιραστεί.

Για ολόκληρο το χρονικό πλαίσιο που έχουμε δημοσιεύσεις από τους χρήστες			
	Για όλους τους χρήστες	Για τους αμοιβαίους φίλους	Για τους μη αμοιβαίους φίλους
Χρήστες	590310	290548	486300
Ενεργοί Χρήστες	473348	172853	418460
Συντελεστής	4.31511791443915E-4	0.001283948075714457	2.3199316040511606E-4

Πίνακας 4.1 Συντελεστής διάδοσης για ολόκληρο το χρονικό διάστημα στο οποίο έχουμε tweets

Έχουμε υπολογίσει το συντελεστή διάδοσης για μία μικρότερη χρονική περίοδο:

Από την 1^η Ιανουαρίου του 2012 μέχρι το τέλος Μαρτίου 2012.

Για όλους τους χρήστες	
Χρήστες	590310
Ενεργοί Χρήστες	375964
Συντελεστής	4.392752052113019E-4

Πίνακας 4.2 Συντελεστής διάδοσης για 1^η Ιανουαρίου του 2012 μέχρι το τέλος Μαρτίου

Για την εβδομάδα με τις περισσότερες δημοσιεύσεις (2 Μαρτίου)			
	Για όλους τους χρήστες	Για τους αμοιβαίους φίλους	Για τους μη αμοιβαίους φίλους
Χρήστες	590310	290548	486300
Ενεργοί Χρήστες	176515	46752	149829
Συντελεστής	0.0013671124691569927	0.0015912320400793426	0.0013194365154875238

Πίνακας 4.3 Συντελεστής διάδοσης για την εβδομάδα με τις περισσότερες δημοσιεύσεις

Βλέπουμε ότι ο συντελεστής διάδοσης κυμαίνεται αρκετά χαμηλά στη γενική περίπτωση 0.04% είναι ακόμα πιο χαμηλός όταν οι χρήστες δεν είναι στενοί φίλοι 0.02 % αλλά αυξάνεται στο 0.12 % όταν οι χρήστες είναι στενοί φίλοι. Αυτό συγκρινό με το συντελεστή στο κοινωνικό δίκτυο του Facebook 0.19% στη γενική περίπτωση βλέπουμε ότι το twitter δεν έχει και τόσο ψηλό ποσοστό επηρεασμού παρόλο που είναι ένα κοινωνικό δίκτυο με στόχο την ενημέρωση και τη διάδοση πληροφοριών αντίθετα με το Facebook που έχει ως σκοπό την επικοινωνία.

Θεωρούμε ότι το αποτέλεσμα που πήραμε για την εβδομάδα που ξεκινά στις 2 Μαΐου είναι συμπτωματική διότι έχουμε βρει το συντελεστή και για άλλα χρονικά διαστήματα με παρόμοιο αριθμό και δεν έχει ξεπεράσει ποτέ το 0.04%

4.2 Συντελεστής Διάδοσης Σχετικά με τις Ζώνες Ώρας

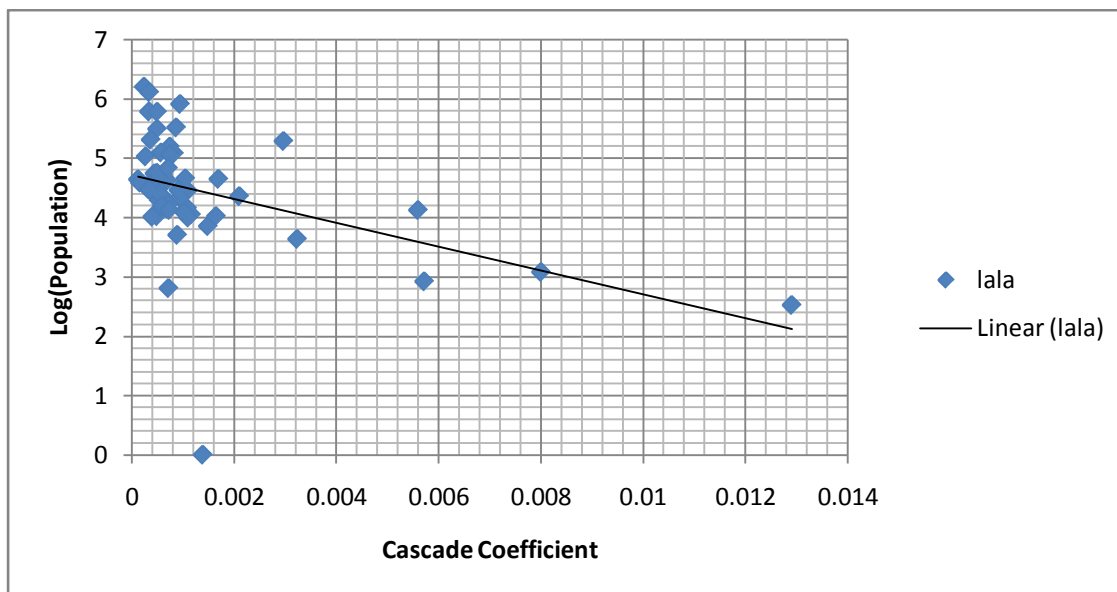
Παίρνοντας ένα χαμηλό συντελεστή διάδοσης θελήσαμε να δούμε σε ποιες περιπτώσεις ο συντελεστής αυτός αυξάνεται. Χρησιμοποιήσαμε τις ώρες ζώνης οι οποίες είναι ένδειξη ότι οι χρήστες βρίσκονται πιο κοντά για να δούμε εάν οι χρήστες επηρεάζονται περισσότερο από τους χρήστες που βρίσκονται πιο κοντά τους.

Το twitter δίνει στον χρήστη την επιλογή να δηλώσει την ζώνη ώρας του. Υπάρχουν 142 ζώνες που μπορούν να επιλεγούν από τους χρήστες. Ο μέσος όρος χρηστών ανά ζώνη είναι 93076.35 ενώ ο μέσος είναι 7167. Η διασπορά είναι ίση με 1.3677E11 και η τυπική απόκλιση είναι ίση με 369823.625. Η μεγάλη διασπορά και τυπική απόκλιση είναι λόγω του μεγάλου μεγέθους που έχουν κάποιες ζώνες. Κάποιες ζώνες εκτείνονται σε συμπλέγματα χωρών ενώ άλλες σε πολύ μικρές πόλεις.

Η μεγαλύτερη τιμή του συντελεστή είναι 10% και είναι στην περιοχή της Astana (Περιοχή στο Καζακστάν με 700 χιλιάδες κατοίκους) ενώ η μικρότερη είναι 0.01%. Ο μέσος όρος τους συντελεστή είναι 0.16% Αυτό δείχνει ότι οι χρήστες επηρεάζονται πολύ περισσότερο από τους χρήστες που είναι πιο κοντά τους. Αξίζει να σημειωθεί ότι από τις 56 περιοχές που παρουσιάζουν διαδόσεις μόνο 10 έχουν πιο χαμηλό συντελεστή διάδοσης από το 0.04% που έχουμε βρει πιο πριν.

Ακολούθως δημιουργήσαμε το γράφημα του συντελεστή διάδοσης σε σχέση με το μέγεθος των ζωνών ώρας, προσπαθώντας να εντοπίσουμε εάν μικρότερες ζώνες έχουν πιο ψηλό συντελεστή διάδοσης.

Είναι προφανές ότι όσο πιο μικρή είναι η περιοχή τόσο πιο ψηλός είναι ο συντελεστής διάδοσης. Αυτό επιβεβαιώνει και ενισχύει την ερεύνα που έγινε στο [4] όπου δήχθηκε ότι οι χρήστες τείνουν να διαδίδουν περιεχόμενο στους πιο κοντινούς τους φίλους.



Σχήμα 4.4 Συντελεστής διάδοσης σε σχέση με το μέγεθος των ζωνών ώρας σε λογαριθμική κλίμακα

4.3 Συντελεστής Διάδοσης Σχετικά με τον Αριθμό των Φίλων που Διαδίδουν

Όσοι περισσότεροι είναι οι φίλοι που διαδίδουν κάποιο αντικείμενο σε ένα χρήστη τότε λογικά τόσο πιο πιθανό θα είναι αυτός ο χρήστης να το υιοθετήσει. Για κάθε εισερχόμενη διάδοση στο χρήστη μετρήσαμε πόσοι φίλοι του έστειλαν το ίδιο αντικείμενο και ποια ήταν η πιθανότητα να μοιραστεί το αντικείμενο ανάλογα με τον αριθμό των φίλων που του το είχαν στείλει.

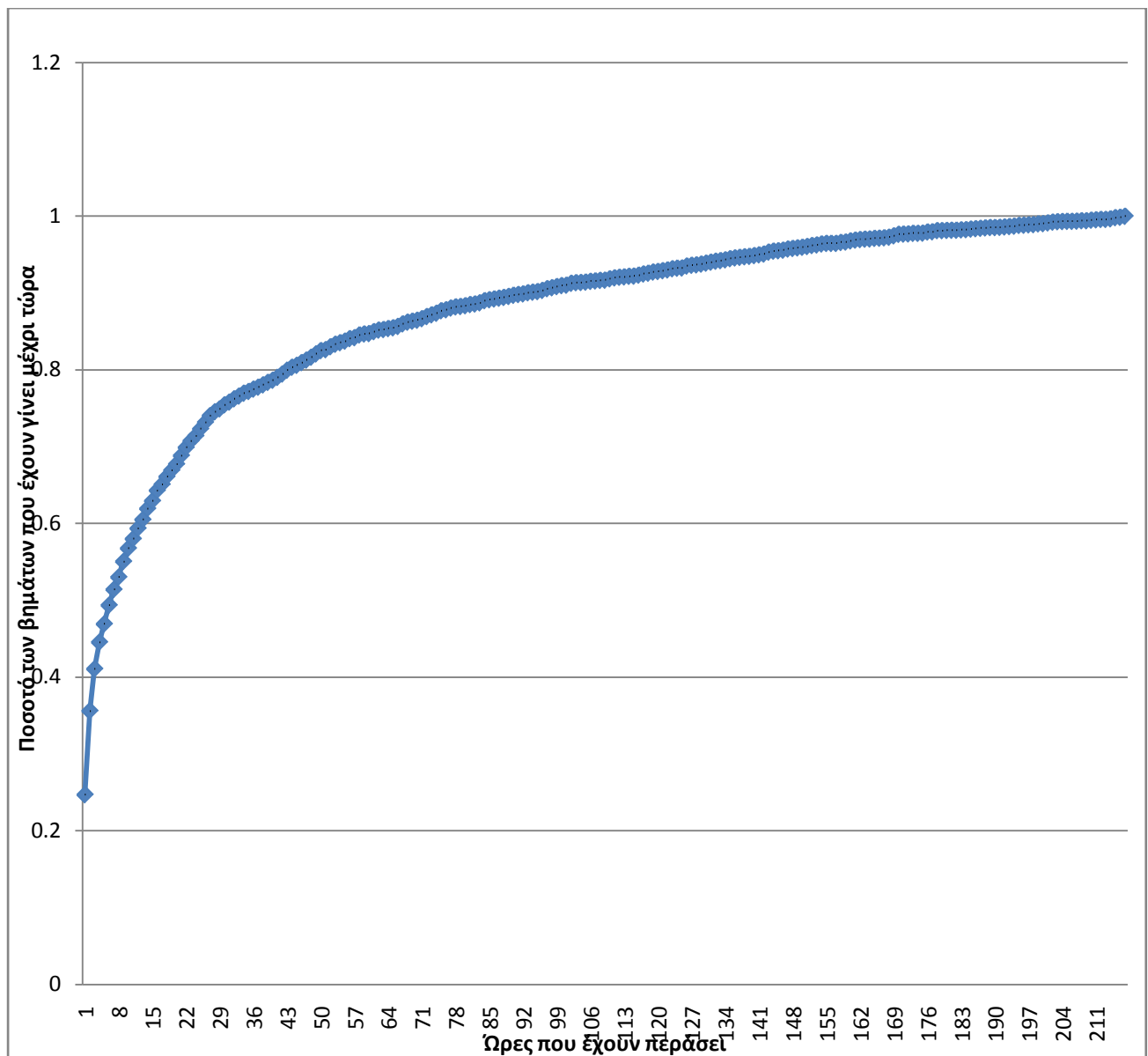


Σχήμα 4.5 Συντελεστής διάδοσης σε σχέση με τον αριθμό των φίλων που μοιράστηκαν το σύνδεσμο

Παρατηρούμε ότι όσοι περισσότεροι φίλοι διαδώσουν σε κάποιο χρήστη ένα αντικείμενο τότε η πιθανότητα να το μοιραστεί αυξάνεται σε μεγάλο βαθμό. Η αύξηση είναι σχεδόν εκθετική όταν ξεπεράσει τον αριθμό των 8 φίλων. Ίσως αυτό να είναι το κατώφλι μετά από το οποίο η εξάπλωση περνά στο στάδιο της πανδημίας και το αντικείμενο γίνεται “Viral”.

4.4 Καθυστέρηση Διάδοσης

Ένα σημαντικό στοιχείο που μελετάται στις κοινωνικές διαδόσεις είναι η διάρκεια. Μετά από πόσο καιρό μία κοινωνική διάδοση παύει να ισχύει. Έχουμε μελετήσει το χρόνο που παίρνει σε ένα χρήστη να διαδώσει το περιεχόμενο που έχει δει μέσα από το κοινωνικό δίκτυο.



Σχήμα 4.6 Ποσοστό των βημάτων διάδοσης που έχουν γίνει ανάλογα με τις ώρες που έχουν περάσει

Όπως βλέπουμε οι διαδόσεις τείνουν να γίνονται πολύ σύντομα μετά που ο χρήστης λαμβάνει το περιεχόμενο. Το 25% των διαδόσεων γίνεται εντός 1 ώρας ενώ στις 3 ώρες έχει γίνει το 40%. Μέχρι το τέλος της πρώτης μέρας έχει γίνει ήδη το 70% των διαδόσεων και στις 3 μέρες το 85%. Την 4^η μέρα γίνονται το 90% των διαδόσεων. Παρόλο που στις επόμενες μέρες συνεχίζουν να υπάρχουν διαδόσεις το ποσοστό είναι αμελητέο και το πιο πιθανό οφείλεται σε άλλα μέσα και όχι σε κοινωνική διάδοση διότι μία δημοσίευση δεν μπορεί να μείνει επαόριστο στην κοινωνική ροή του χρήστη. Η δεύτερη εξήγηση είναι ότι το περιεχόμενο έχει κάνει κύκλο και έχει ξαναφτάσει στους χρήστες δίνοντας στους χρήστες που δεν το είχαν μοιραστεί αρχικά να το μοιραστούν.

4.5 Σχέση Μεταξύ Κοινωνικής Διάδοσης και Δημοτικότητας

Ποια είναι η σχέση που έχει ο αριθμός των κοινωνικών διαδόσεων ενός βίντεο με τις πραγματικές θεάσεις που έχει. Ίσως ένα βίντεο να μπορεί να έχει ψηλές θεάσεις και δημοτικότητα ενώ δεν διαδίδεται στο κοινωνικό δίκτυο.

Χωρίσαμε τα βίντεο σε κατηγορίες ανάλογα με τον αριθμό των θεάσεων τους και ακολούθως μετρήσαμε τις κοινωνικές διαδόσεις ανά κατηγορία

Κατηγορία (θεάσεις)	Αριθμός Βίντεο στην κατηγορία	Συνολικές διαδόσεις	Μέσος όρος διαδόσεων
1 000	68655	85252	1.241744956667395
5 000	37899	45640	1.2042534103802212
20 000	43014	49197	1.1437438973357512
50 000	34855	40515	1.1623870319896714
200 000	53509	68099	1.2726644115943113
400 000	23544	34638	1.4712028542303772
700 000	16010	27011	1.6871330418488444
1 εκατ.	8571	16452	1.91949597479874
2 εκατ.	13332	27721	2.0792829282928293
5 εκατ.	11183	30245	2.7045515514620404
10 εκατ.	4641	18924	4.077569489334195
20 εκατ.	2205	12967	5.880725623582767
50 εκατ.	1210	10775	8.90495867768595
100 εκατ.	393	10519	26.765903307888042
200 εκατ.	21	553	26.333333333333332

350 εκατ.	7	176	25.142857142857142
-----------	---	-----	--------------------

Πίνακας 4.7 Κοινωνικές διαδόσεις σε σχέση με τον αριθμό των θεάσεων

Είναι προφανές ότι υπάρχει κάποια σχέση μεταξύ των θεάσεων και των διαδόσεων. Όσο πιο πολλές είναι οι θεάσεις του βίντεο τόσες περισσότερες είναι και οι κοινωνικές διαδόσεις. Στον πιο πάνω πίνακα παρατηρούμε 2 απότομες αυξήσεις στον αριθμό των διαδόσεων. Στα πρώτα βίντεο βλέπουμε τον αριθμό των διαδόσεων να αυξάνεται λίγο σε σχέση με τη μεγάλη αύξηση του αριθμού των θεάσεων ενώ στις 5 εκατομμύρια θεάσεις από τις 2 έχουμε μία αύξηση του 140%. Η δεύτερη απότομη αύξηση είναι από τα 50 εκατομμύρια στα 100 εκατομμύρια όπου ο μέσος όρος διαδόσεων τριπλασιάζεται. Ο μέσος όρος θεάσεων μένει σταθερός για βίντεο με περισσότερες από 100 εκατομμύρια θεάσεις. Η μεγάλη αύξηση του αριθμού των διαδόσεων ενώ το ποσοστό των θεάσεων δεν έχει αυξηθεί σε τόσο μεγάλο βαθμό ίσως να δείχνει ότι οι κοινωνικές διαδόσεις να είναι αυτές που φέρνουν στα βίντεο τις υπερβολικά μεγάλες θεάσεις.

Κεφάλαιο 5

Γεννήτρια Αιτήσεων για το YouTube

5.1 Εισαγωγή	33
5.2 Κοινωνικές πληροφορίες	33
5.3 Πληροφορίες αιτήσεων	34
5.4 Αλγόριθμος	36

5.1 Εισαγωγή

Για την αξιολόγηση κάποιου μοντέλου προσωρινής αποθήκευσης είναι αναγκαίο να δημιουργήσουμε τις αιτήσεις με τις οποίες θα δοκιμάσουμε το μοντέλο. Αυτές οι αιτήσεις πρέπει να είναι όσο πιο κοντά στις πραγματικές αιτήσεις των χρηστών γίνεται. Όλα τα προηγούμενα συστήματα υπολείπονταν στην αξιολόγηση του μοντέλου τους. Τα πλείστα συστήματα δήλωναν ότι ο χρήστης θα ζητήσει ένα αντικείμενο το οποίο συνάντησε σε ένα κοινωνικό δίκτυο με αυθαίρετη πιθανότητα p .

Ακόμα δεν λαμβάνονταν υπόψη οι διαφορετικοί χρόνοι στους οποίους οι χρήστες ζητούν περιεχόμενο, όπως για παράδειγμα οι χρήστες κάνουν περισσότερες αιτήσεις στις ώρες αιχμής και οι ώρες αιχμής είναι διαφορετικές για χρήστες που βρίσκονται σε διαφορετικές χρονικές ζώνες.

Έχουμε δημιουργήσει μια γεννήτρια αιτήσεων για τη σελίδα του YouTube η οποία λαμβάνει υπόψη όλες τις δυνατές παραμέτρους και είναι ανεξάρτητη του κοινωνικού δικτύου από το οποίο αντλούμε τις πληροφορίες. Η γεννήτρια αυτή μπορεί να παραμετροποιηθεί για οποιαδήποτε σελίδα και ακόμα και αν το περιεχόμενο της δεν είναι πολυμεσικό.

5.2 Κοινωνικές Πληροφορίες

Βασικοί συντελεστές στην διάδοση πληροφοριών στο διαδίκτυο είναι τα διάφορα κοινωνικά δίκτυα. Ένας χρήστης μοιράζεται περιεχόμενο το οποίο είναι ορατό σε ένα υποσύνολο των κόμβων του κοινωνικού δικτύου (φίλους, οπαδούς, οι συνδρομητές) και μερικοί από αυτούς τους χρήστες να μοιράζονται περιεχόμενο επίσης με τις συνδέσεις τους, διαδίδοντας το μέσω

του κοινωνικού δικτύου. Η ισχύς των δεσμών μεταξύ των χρηστών ποικίλει σε ισχύ, ένα υποσύνολο των χρηστών είναι πιο πιθανό να δει το περιεχόμενο που μοιράζεται από κάποιο χρήστη ότι το υπόλοιπο τις υπόλοιπες γνωριμίες του, αλλά όπως φαίνεται στο [26], οι πιο άφθονοι οι ασθενείς δεσμοί είναι η αυτοί που είναι υπεύθυνοι για τη μετάδοση των πληροφοριών, αν και οι ισχυρότεροι δεσμοί έχουν ξεχωριστά μεγαλύτερη επιρροή. Έτσι, με την απλή εξέταση του κοινωνικού γράφου, και τις τοποθεσίες των χρηστών μπορούμε να εκτιμήσουμε την εξάπλωση ενός αντικειμένου.

Η ισχύς της κοινωνικής διάδοσης αναλύεται στο [26], και εκτιμάται ότι οι χρήστες που βλέπουν ένα σύνδεσμο στο κοινωνικό δίκτυο είναι 7,35 φορές πιο πιθανό να μοιραστούν το συγκεκριμένο σύνδεσμο από τους χρήστες που δεν τον βλέπουν στο κοινωνικό δίκτυο. Αυτό έχει υπολογιστεί κάνοντας ένα πείραμα μεγάλης κλίμακας στο κοινωνικό δίκτυο του facebook. Για κάθε σύνδεσμο που μοιράζεται στο κοινωνικό δίκτυο επιλέγονται κάποιοι χρήστες από τους οποίους το περιεχόμενο αποκρύβεται και δεν φαίνεται στην κοινωνική τους ροή. Έτσι δημιουργούνται δύο ξεχωριστές περιπτώσεις το feed και το no feed. Ακολούθως υπολογίζεται η πιθανότητα (likelihood) ένας χρήστης να μοιραστεί περιεχόμενο σε κάθε περίπτωση και υπολογίζεται το πηλίκο το οποίο δείχνει πόσο πιο πιθανό είναι ένας χρήστης να μοιραστεί κάτι που έχει δει στο κοινωνικό σε σχέση με την περίπτωση που δεν το έβλεπε. Η πιθανότητα να μοιραστεί κάποιος στην περίπτωση feed είναι 0.191% διαιρούμενη με να μοιραστεί κάποιος στην περίπτωση no feed 0.025% δείχνει ότι ένα άτομο είναι 7,37 φορές πιο πιθανό να μοιραστεί κάτι στην περίπτωση feed (95% CI = [7:23; 7:72]).

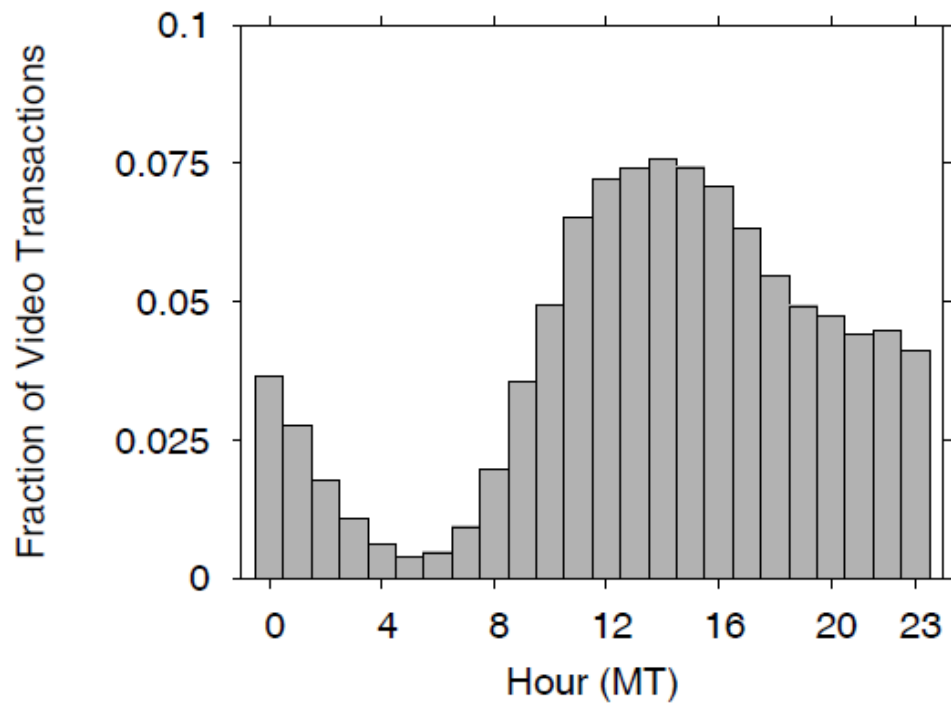
Αν λάβουμε υπόψη τον αριθμό των χρηστών που απλά βλέπουν το σύνδεσμο ο αριθμός είναι πολύ υψηλότερος. Επίσης, ένα στους 12,5 συνδέσμους ξαναμοιράζεται και αυτό είναι πολύ λογικό λαμβάνοντας υπόψη το μεγάλο ποσό των γνωριμιών που έχει κάθε χρήστης.

5.3 Πληροφορίες Αιτήσεων

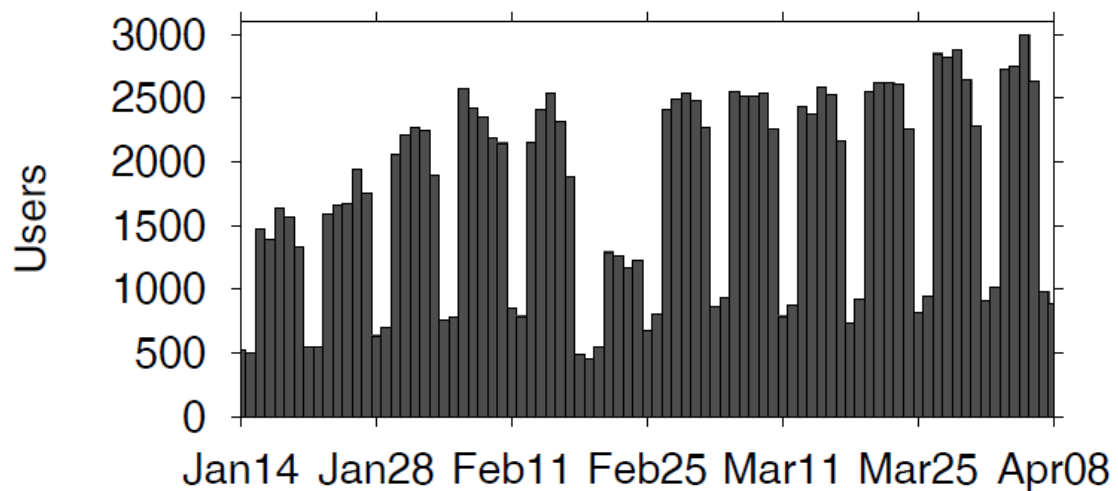
Οι αιτήσεις των χρηστών ακολουθούν ημερήσιες τάσεις. Δηλαδή σε κάποιες ώρες της μέρας γίνονται πολύ περισσότερες αιτήσεις από άλλες ώρες και ακόμα οι αιτήσεις διαφοροποιούνται ανάλογα με τη μέρα της εβδομάδας.

Στο [8] δίνονται οι κατανομές των αιτήσεων ανάλογα με τη μέρα της εβδομάδας και την ώρα της μέρα. Όπως βλέπουμε ο αριθμός των αιτήσεων αυξάνεται κατά τη διάρκεια της εβδομάδας και μειώνεται δραματικά τα σαββατοκύριακα. Εξαιρέσεις σε αυτή την κατανομή αποτελούν κάποιες μέρες της εβδομάδας οι οποίες είναι αργίες και οι οποίες επιδεικνύουν παρόμοια συμπεριφορά με τα σαββατοκύριακα. Σχετικά με την ώρα της μέρας βλέπουμε ότι

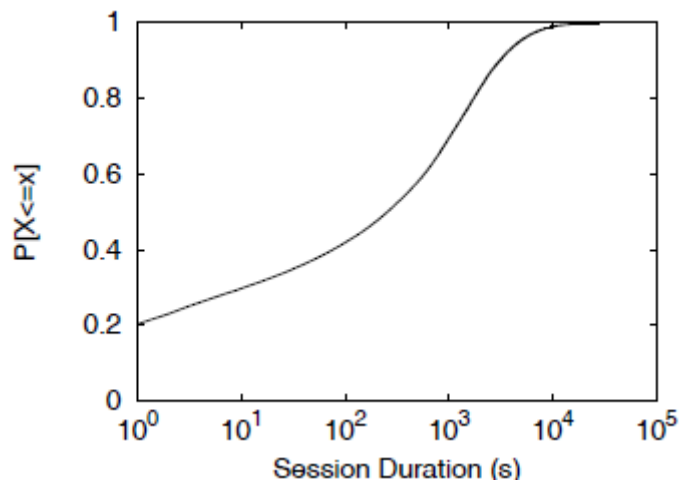
επιβεβαιώνεται η αντίληψη ότι οι αιτήσεις αυξάνονται και κορυφώνονται κατά το μεσημέρι ενώ είναι ψηλές και τις νυχτερινές ώρες ενώ είναι μηδαμινές κατά τις μεταμεσονύκτιες ώρες.



Σχήμα 5.1 Κατανομή των αιτήσεων στις ώρες της μέρας [8]



Σχήμα 5.2 Κατανομή των αιτήσεων στις διάφορες μέρες της εβδομάδας[8]



Σχήμα 5.3 Η αθροιστική συνάρτηση κατανομής για τη διάρκεια κάθε συνεδρίας. [30]

5.4 Αλγόριθμος

Χρησιμοποιούμε την τεχνική που παρουσιάζεται στο [31] για την επιλογή των βίντεο για κάθε αίτηση επεκτείνοντας την για να χρησιμοποιεί και κοινωνικές πληροφορίες για τους χρήστες. Οι αιτήσεις για κάθε βίντεο πρέπει να σχετίζονται με τη δημοτικότητα του κάθε βίντεο. Έτσι χρησιμοποιούμε τον αριθμό θεάσεων του κάθε βίντεο για να επιλέξουμε βίντεο για κάθε αίτηση. Δημιουργούμε ένα πίνακα Σ με τα αθροίσματα των θεάσεων όλων των βίντεο, όπου στη θέση i του πίνακα βρίσκεται το άθροισμα όλων των βίντεο με ταυτότητα μικρότερη ή ίση του i . Ακολουθώς δημιουργούμε ένα τυχαίο αριθμό κ μεταξύ του 0 και του συνολικού αθροίσματος και επιλέγουμε το βίντεο j όπου $\Sigma[j-1] \leq \kappa < \Sigma[j]$. Έτσι όλα τα βίντεο έχουν πιθανότητα να επιλεγθούν και η πιθανότητα που έχουν για να επιλεγθούν είναι ανάλογη της δημοτικότητάς τους. Η επιλογή του βίντεο γίνεται με δυαδική επιλογή έτσι ο χρόνος που χρειάζεται να επιλεγεί το κάθε βίντεο είναι λογαριθμικός σε σχέση με τον αριθμό των βίντεο.

Στην περίπτωση που ο χρήστης είναι αποδέκτης κοινωνικών διαδόσεων για κάποια βίντεο αυξάνουμε την πιθανότητα να επιλεγθούν αυτά τα βίντεο κατά ένα πραγματικό αριθμό K που δίνεται ως παράμετρος, πολλαπλασιάζοντας τις θεάσεις τους επί το K και ακολουθώς χρησιμοποιούμε την πιο πάνω τεχνική. Ο αριθμός K δίνει το πόσο πιο πιθανόν είναι ένας χρήστης να δει κάποιο βίντεο το οποίο έχει βρει στο κοινωνικό δίκτυο παρά να το δει χωρίς το κοινωνικό δίκτυο.

Δεδομένα εισόδου – εξόδου της γεννήτριας αιτήσεων:

Είσοδος:

- **Βασική είσοδος**

Πραγματική ημερομηνία και την ώρα έναρξης της προσομοίωσης

Οι ώρες για τις οποίες η προσομοίωση θα τρέχει

Οι χρήστες και η ζώνη ώρας του κάθε χρήστη

Τα βίντεο με τον αριθμό θεάσεων και η διάρκεια του κάθε βίντεο

- **Κατανομές**

Η κατανομή του πόσοι χρήστες χρησιμοποιούν την υπηρεσία ανά ημέρα της εβδομάδας

Η κατανομή για το πώς οι συνεδριάσεις κατανέμονται μέσα στην ημέρα

Η αθροιστική συνάρτηση κατανομής (CDF) της διάρκειας των συνόδων

- **Κοινωνικές πληροφορίες (όχι υποχρεωτικά)**

Για κάθε χρήστη οι εισερχόμενες διαδόσεις από το κοινωνικό δίκτυο με μια χρονοσφραγίδα (Τα βίντεο που μοιράστηκαν προς τον χρήστη)

Για κάθε χρήστη οι εξερχόμενες διαδόσεις από το κοινωνικό δίκτυο με μια χρονοσφραγίδα (Τα βίντεο που μοιράστηκαν από το χρήστη)

Έξοδος:

Τα αιτήματα των χρηστών με τη μορφή:

Ταυτότητα Βίντεο

Ταυτότητα χρήστη

Χρονοσφραγίδα

Όπου η χρονοσφραγίδα μετριέται σε δευτερόλεπτα από την έναρξη της προσομοίωσης

Αλγόριθμος:

Βήμα 1: Διάβασε όλες τις κατανομές

Βήμα 2: Διάβασε όλα τα βίντεο

Βήμα 3: Διάβασε όλους τους χρήστες

Βήμα 4: Διάβασε τη ζώνη ώρας, τη χρονική στιγμή που ξεκινά η προσομοίωση σε ώρα UTC, τη χρονική διάρκεια της προσομοίωσης και τον αριθμό K

Βήμα 5: Για κάθε χρήστη διάβασε τις εισερχόμενες και εξερχόμενες διαδόσεις (cascades)

Βήμα 6: Δημιούργησε ένα πίνακα από λίστες *Active* 100,000 θέσεων ο οποίος θα περιέχει τις ενεργές συνεδριάσεις για κάθε δευτερόλεπτο που έρχεται.

Βήμα 7: Για κάθε ώρα που τρέχει η προσομοίωση:

Βήμα 7.1 Σύμφωνα με την ώρα και τη μέρα δημιούργησε αριθμό συνεδριών σύμφωνα με τις κατανομές ώρας και μέρας, και θέσε για κάθε συνεδρία διάρκεια σύμφωνα με την κατανομή της διάρκειας κάθε συνεδρίας, ανάθεσε της τυχαία ένα χρήστη και θέσε ως ώρα έναρξης s ένα αριθμό από το 0 μέχρι το 3599 (θα ξεκινά μέσα στην ώρα) τοποθετώντας την στην αντίστοιχη λίστα του πίνακα *Active*

Βήμα 7.2 Για κάθε δευτερόλεπτο i στην ώρα από 0 μέχρι 3599

Βήμα 7.2.1 Για κάθε συνεδρία που ξεκινά ή συνεχίζει σε αυτό το δευτερόλεπτο (Βρίσκεται στην αντίστοιχη λίστα του πίνακα *Active[i]*)

Βήμα 7.2.1.1 Επέλεξε ένα βίντεο για να δει ο χρήστης με τον τρόπο που περιγράφηκε πιο πάνω και τύπωσε το στην έξοδο

Βήμα 7.2.1.1 Μείωσε τον χρόνο της συνεδρίας όση είναι η διάρκεια του βίντεο που επιλέχθηκε και αν μένει ακόμα χρόνος στη συνεδρία τοποθέτησε την να συνεχίζει όταν τελειώσει το βίντεο. Στη θέση του πίνακα *active* $i + \text{διάρκεια βίντεο}$

Βήμα 7.3 Μετακίνησε κάθε λίστα στον πίνακα *Active* κατά 3600 θέσεις πίσω.

Για κάθε ώρα που τρέχει η προσομοίωση δημιουργούνται οι αιτήσεις σύμφωνα με τις κατανομές, και για κάθε αίτηση επιλέγεται τυχαία ένα βίντεο ανάλογα με τη δημοτικότητα του και τις κοινωνικές διαδόσεις που επηρεάζουν το χρήστη.

Πολυπλοκότητα:

Έστω ότι έχουμε n χρήστες, m βίντεο, s συνεδρίες, r αιτήσεις, h ώρες προσομοίωσης :

Βήμα 1 : $O(1)$

Βήμα 2: $O(n)$

Βήμα 3: $O(m)$

Βήμα 4: $O(1)$

Βήμα 5: $O(n * \text{constant})$ Ένας χρήστης δεν μπορεί να έχει υπερβολικά μεγάλο αριθμό διαδόσεων

Βήμα 6: $O(1)$

Βήμα 7.1 και 7.2: Βλέπουμε ότι σε κάθε κύκλο (loop) αυτών των βημάτων δημιουργείται ένας αριθμός αιτήσεων. Ο συνολικός αριθμός κύκλων που μπορεί να γίνουν είναι $h*3600$. Ο αριθμός των αιτήσεων που γίνονται είναι πολύ μεγαλύτερος από τον αριθμό των ωρών και άρα $T=O(r) * \text{κόστος για κάθε αίτηση} = \log(m)$
 $T=O(r * \log(m))$

Βήμα 7.3: $O(1)$

Συνολικός χρόνος εκτέλεσης : $O(n + m + r * \log(m))$

Όπου το r είναι της τάξεως του αριθμού των χρηστών n

Άρα η πολυπλοκότητα είναι : $O(n * \log(m))$

Κεφάλαιο 6

Εφαρμογές

6.1 Προβλήματα στα Δίκτυα Παράδοσης Περιεχομένου	40
6.2 Πολυμέσα	41
6.3 Κοινωνικά δίκτυα	42
6.4 Διατύπωση προβλήματος	44

Η ανάλυση των κοινωνικών διαδόσεων που έχει γίνει έχει διάφορες εφαρμογές. Μία άμεση εφαρμογή είναι η υιοθέτηση των συμπερασμάτων που έχουμε εξάγει για τη βελτίωση των πολιτικών που εφαρμόζονται στα Δίκτυα Παράδοσης Περιεχομένου. Επίσης η γεννήτρια αιτήσεων είναι απαραίτητη για οποιαδήποτε δοκιμή και έλεγχο κάποιου μοντέλου που να υιοθετεί κάποια πολιτική προσωρινής αποθήκευσης.

6.1 Προβλήματα στα Δίκτυα Παράδοσης Περιεχομένου

Οι στρατηγικές που ακολουθούνται μέχρι τώρα στα δίκτυα παράδοσης περιεχομένου δεν μπορούν να χειριστούν τα ολοένα και μεγαλύτερα και περισσότερα ιστιακά αντικείμενα που δημιουργούνται. Για παράδειγμα η δημοτικότητα των βίντεο ακολουθεί long-tail κατανομή όπου ένα πολύ μικρό ποσοστό των βίντεο είναι υπεύθυνο για τις περισσότερες αιτήσεις. Η στρατηγική που ακολουθείται μέχρι τώρα θα μπορούσε να χειριστεί τα πολύ δημοφιλή αντικείμενα αντιγράφοντας τα σε όλες τις τοποθεσίες. Τα μη δημοφιλή αντικείμενα δεν θα έμεναν για αρκετό χρόνο αποθηκευμένα στους υποκατάστατους εξυπηρετητές διότι λόγω του μεγάλου τους αριθμού θα εναλλάσσονταν συνέχεια. Οπότεν κάθε φορά που ένα μη δημοφιλή αντικείμενο ζητείται από ένα χρήστη θα μεταφέρεται από τον πηγαίο εξυπηρετητή αυξάνοντας το φόρτο εργασίας του.

Για να επιλύσουμε αυτό το πρόβλημα η διαίσθησή μας λέει ότι πρέπει να ανακαλύψουμε ποια βίντεο και άλλα μεγάλου μεγέθους αντικείμενα θα γίνουν δημοφιλή παγκόσμια και θα αντιγραφούν σε όλες τις τοποθεσίες και ποια θα γίνουν δημοφιλή σε συγκεκριμένες τοποθεσίες και να αντιγραφούν μόνο σε αυτές τις τοποθεσίες. Προσπαθούμε να

προσεγγίσουμε αυτό το πρόβλημα από τη σκοπιά του κοινωνικού δικτύου, χρησιμοποιούμε δεδομένα από τα κοινωνικά δίκτυα για να εξάγουμε τις τάσεις των μελλοντικών αιτήσεων για πολυμεσικά αντικείμενα και να τοποθετήσουμε τα αντικείμενα κοντά στους χρήστες που θα τα ζητήσουν. Όταν ένας χρήστης μοιράζει κάποιο περιεχόμενο σε ένα κοινωνικό δίκτυο και ένας άλλος χρήστης φίλος του πρώτου το ξαναμοιράζει τότε λέμε ότι έχουμε κοινωνική διάδοση. Οι κοινωνικές διαδόσεις (Cascades) δίνουν πληροφορίες για τις τάσεις των χρηστών στα κοινωνικά δίκτυα.

6.2 Πολυμέσα

Το video on demand (VOD) και το περιεχόμενο δημιουργημένο από τους ίδιους τους χρήστες είναι υπεύθυνα για ένα πολύ μεγάλο ποσοστό της κίνησης στο διαδίκτυο. 15-25% όλης της κίνησης στο διαδίκτυο προκαλείται μόνο από βίντεο. Ο μεγαλύτερος παροχέας βίντεο είναι η ιστοσελίδα YouTube η στην οποία 2 δισεκατομμύρια βίντεο παρακολουθούνται καθημερινά και εκατοντάδες χιλιάδες βίντεο γίνονται καθημερινά upload [11]. Η δημοτικότητα των βίντεο στο YouTube ακολουθεί long-tail κατανομή όπου 10% των βίντεο είναι υπεύθυνα για το 80% των παρακολουθήσεων και τα υπόλοιπα ζητούνται πολύ λίγες φορές [5]. Η αρχή Pareto (κανόνας του 80-20) παρατηρείται και στο [7] όπου βρίσκεται ότι σε μία δημοφιλή VOD υπηρεσία το 10% των πιο δημοφιλή αντικειμένων είναι υπεύθυνο για το 60% των αιτήσεων. Αυτό ενδυναμώνει την πεποίθησή μας ότι τα παραδοσιακά συστήματα παράδοσης περιεχομένου δεν μπορούν να αντεπεξέλθουν με τα μη δημοφιλή αντικείμενα και επίσης έχουν υπερβολικές ανάγκες σε bandwidth για να τύχουν χειρισμού από την απλή αρχιτεκτονική πελάτη – εξυπηρετητή.

Επίσης οι αιτήσεις στο YouTube ακολουθούν ημερήσιες τάσεις (diurnal trends) [7,11]. Δηλαδή οι αιτήσεις ακολουθούν ξεκάθαρα ένα μοτίβο ημέρας-νύχτας όπου οι περισσότερες αιτήσεις γίνονται γύρω στο μεσημέρι και νωρίς το βράδυ ενώ στις άλλες ώρες ο αριθμός των αιτήσεων περιορίζεται. Επίσης ο ρυθμός με τον οποίο γίνονται οι αιτήσεις διαφοροποιείται στις αργίες. Οι αιτήσεις μειώνονται δραματικά αλλά μόνο μετά που περνούν οι 2-3 πρώτες μέρες. Αυτές οι τάσεις μπορεί να χρησιμοποιηθούν από το σύστημα για να γίνουν και μεταφορές δεδομένων στις ώρες που δεν υπάρχει πολλή κίνηση στο σύστημα. Επίσης οι μέρες αργιών μπορούν να χρησιμοποιηθούν για να γίνονται ενημερώσεις και επιδιορθώσεις στο σύστημα.

6.3 Κοινωνικά δίκτυα:

Τα κοινωνικά δίκτυα έχουν επανακαθορίσει το διαδίκτυο αλλάζοντας το επίκεντρο από το περιεχόμενο στους χρήστες. Οι χρήστες συνδέονται με τους χρήστες που γνωρίζουν στην πραγματική ζωή, όπως η οικογένεια φίλοι και συνάδελφοι, χρήστες που έχουν τα ίδια ενδιαφέροντα, διάσημους ανθρώπους ή απλά τυχαίους χρήστες που δεν έχουν καμία σχέση μαζί τους. Είναι δυνατή η αποστολή μηνυμάτων, περιεχομένου, της θέσης που βρίσκεται ο χρήστης, σχολιασμός σε δημοσιεύσεις άλλων χρηστών. Μπορούν να ακολουθήσουν κάποιο χρήστη και να ενημερωθούν σχετικά με τη γεωγραφική θέση, τα μηνύματα και να συζητήσουν μαζί του.

Αυτό το τεράστιο ποσό των πληροφοριών, συμπεριλαμβανομένων των σχέσεων μεταξύ των χρηστών μπορεί να χρησιμοποιηθεί με πολλούς τρόπους. Οι καμπάνιες μάρκετινγκ που βασίζονται στην πλατφόρμα του Facebook; Προεδρικές εκστρατείες των υποψηφίων στο YouTube; Ηθοποιοί που κάνουν τουιτ (tweet) την καθημερινή ρουτίνα τους για να κρατήσουν το ενδιαφέρον του κοινού σε υψηλά επίπεδα. Τα κοινωνικά δίκτυα μπορούν να χρησιμοποιηθούν για την αύξηση της αποτελεσματικότητας των τεχνικών για την προσωρινή αποθήκευση πολυμέσων, όπως βίντεο και εικόνες. Μπορούμε να ανακαλύψουμε ποια αντικείμενα στο Παγκόσμιο Ιστό θα καταστούν δημοφιλή και όπου, φέρνοντας πιο κοντά το περιεχόμενο του χρήστη θα μειώσουμε το χρόνο απόκρισης και τη συμφόρηση δικτύου.

Τα κοινωνικά δίκτυα έχουν ενδιαφέρουσες και χρήσιμες μακροσκοπικές και μικροσκοπικές ιδιότητες [6]. Ο κοινωνικός γράφος παρουσιάζει power-law κατανομή, όπως το παραδοσιακό ιστό, με τη διαφορά ότι η εισερχόμενος και εξερχόμενος βαθμός των χρηστών στα κοινωνικά δίκτυα έχουν υψηλό βαθμό συσχέτισης, όταν στον παγκόσμιο ιστό το ποσό των κόμβων με υψηλότερο βαθμό εξερχόμενων συνδέσεων είναι πολύ μεγαλύτερο. Η υψηλή συσχέτιση παράγεται από τον υψηλό αριθμό των συμμετρικών συνδέσεων. Η συμμετρία των συνδέσεων θα μπορούσε να οφείλεται στην τάση των χρηστών να ανταποδώσει συνδέσμους που οδηγούν σε αυτούς και την αρχιτεκτονική των κοινωνικών δικτύων με μη κατευθυνόμενες συνδέσεις. Υψηλή συσχέτιση εισερχόμενου και εξερχόμενου βαθμού του κόμβου, κάνει πιο δύσκολη την ανακάλυψη των πιο δημοφιλών και σημαντικών χρηστών.

Το joint degree distribution είναι μία μετρική που δείχνει πόσο συχνά κόμβοι με διαφορετικούς βαθμούς συνδέονται μεταξύ τους και είναι ένα μέτρο για τις δυνατότητες της δομής του δικτύου για την παραγωγή επιδημιών. Στα δίκτυα στα οποία κόμβοι με υψηλό βαθμό συνδέονται με κόμβους με υψηλό βαθμό είναι πιο πιθανό να γίνουν επιδημίες από τα δίκτυα όπου οι κόμβοι με υψηλό βαθμό συνδέονται με κόμβους με χαμηλό βαθμό. Έχει παρατηρηθεί ότι οι κόμβοι με υψηλό βαθμό συνδέονται με κόμβους με υψηλό βαθμό με την εξαίρεση του YouTube που έχει μια πιο "οδηγούμενη από διασημότητες φύση». Από την JDD μπορούμε να υπολογίσουμε επίσης την scale-free metric των δικτύων, που είναι υψηλή στα κοινωνικά δίκτυα που έχουν μελετηθεί και πάλι με την εξαίρεση του YouTube.

Το γεγονός ότι τα μεγάλα σύνολα δεδομένων αν και χρήσιμα μπορεί να είναι παραπλανητικά παρουσιάζεται στο [7], όπου η ανάλυση των κοινωνικών δικτύων γίνεται με τα πραγματικά αιτήματα HTTP και τις απαντήσεις μεταξύ των χρηστών και του Facebook. Αυτή η clickstream ανάλυση έδειξε ότι ο βαθμός γράφημα μπορεί να είναι παραπλανητικός την εξέταση της κοινωνικής διάδοσης πληροφοριών. Παρά τον υψηλό βαθμό στους χρήστες του δικτύου στο γράφο τείνουν να βλέπουν πολύ λίγα προφίλ. Σε αυτό προστίθεται το γεγονός ότι παρόλο που ο μέσος αριθμός των προφίλ που επισκέπτονται ανά συνεδρία είναι 6 ο αριθμός των μοναδικών προφίλ είναι 3. Αυτό σημαίνει ότι οι χρήστες δεν είναι πιθανό να δουν το περιεχόμενο από όλες τις επαφές τους. Για να δούμε αυτό το γεγονός με ωφέλιμο τρόπο με τον οποίο πρέπει επίσης να λάβουμε υπόψη ότι τα πλείστα κοινωνικά δίκτυα εμφανίζουν στην πρώτη σελίδα μία κοινωνική ροή (social feed) που προβάλλεται κατά τη σύνδεση του χρήστη όπου εμφανίζονται οι πρόσφατες ενημερώσεις από τις επαφές του χρήστη. Αυτό διαφοροποιεί και την προοπτική για κάθε κοινωνικό δίκτυο.

6.4 Διατύπωση Προβλήματος:

Θέλουμε να ορίσουμε μια πολιτική αντιγραφής αντικειμένων έτσι ώστε να ελαχιστοποιήσουμε τον χρόνο απόκρισης λαμβάνοντας υπόψη τους περιορισμούς από το μέγεθος αποθηκευτικού χώρους του κάθε εξυπηρετητή, την τοποθεσία του κάθε εξυπηρετητή και την τοπολογία του δικτύου. Προσεγγίζουμε το ζήτημα χρησιμοποιώντας δεδομένα από κοινωνικά δίκτυα (user's posts, tagged content etc.) για να εντοπίσουμε αντικείμενα τα οποία γίνονται δημοφιλή σε μία ή περισσότερες περιοχές και να αντιγράψουμε αυτά τα αντικείμενα στους υποκατάστατους εξυπηρετητές αυτής της περιοχής.

- $G(V,E)$: Γράφος που αναπαριστά το κοινωνικό δίκτυο
 $V=\{V_0,V_1...V_n\}$ οι κόμβοι που αναπαριστούν τους χρήστες του κοινωνικού δικτύου
 $E=\{E_{00}, E_{01}..E_{0n}..E_{nn}\}$ οι ακμές που αναπαριστούν τις συνδέσεις μεταξύ των χρηστών
όπου η ακμή E_{ij} αναπαριστά ότι ο κόμβος i είναι φίλος με τον κόμβο j
- R : Σύνολο περιοχών
 $R=\{r_1,r_2...r_t\}$ $r_i \subset V$ όπου κάθε περιοχή περιέχει ένα αριθμό κόμβων
- N : Σύνολο υποκατάστατων εξυπηρετητών
 $N=\{n_1,n_2..n_u\}$ Ο κάθε υποκατάστατος εξυπηρετητής ανήκει σε μια περιοχή r_i
- C_i : Χωρητικότητα του υποκατάστατου εξυπηρετητή i , $i \in N$ σε bytes
- O : Σύνολο αντικειμένων
 $O = \{o_1,o_2...o_w\}$ δηλώνουν αντικείμενα που οι χρήστες μπορούν να ζητούν και να μοιράζονται
- S_i : Μέγεθος του αντικειμένου i , $i \in O$
- P : Οι δημοσιεύσεις των χρηστών στο κοινωνικό δίκτυο
 $P=\{p_{00},p_{01}...p_{nw}\}$ όπου p_{ij} δηλώνει ότι ο κόμβος i έχει μοιραστεί το αντικείμενο j
στο κοινωνικό δίκτυο

- Q : Αιτήσεις των αντικειμένων από την ιστοσελίδα που περιέχει τα πολυμεσικά αντικείμενα
 $Q=\{q_1,q_2,...q_z\}$ όπου q_i δηλώνει μια αίτηση για ένα αντικείμενο από το σύνολο O

Σκοπός:

Πρέπει να προτείνουμε μια πολιτική η οποία δοθέντος ενός κοινωνικού γράφου $G(V,E)$ και ένα σύνολο από περιοχές R οι οποίες διαχωρίζουν τους κόμβους του κοινωνικού δικτύου και τα ποστ P των κόμβων να αναγνωρίζει τα αντικείμενα του συνόλου O τα οποία θα γίνουν δημοφιλή μόνο σε κάποιες περιοχές έτσι ώστε να γνωρίζουμε πο υ θα αντιγράψουμε περιεχόμενο.

Έστω ότι δημιουργούμε μια συνάρτηση $Predict(G,P, O,R)$ η οποία δοθέντος του κοινωνικού γράφου τα ποστ και την κατανομή των χρηστών στις περιοχές προβλέπει ποια αντικείμενα θα γίνουν παγκόσμια δημοφιλή και ποια θα γίνουν τοπικά δημοφιλή και σε ποιες περιοχές.

Χρησιμοποιώντας το σύνολο των δημοφιλή αντικειμένων και το σύνολο των υποκατάστατων εξυπηρετητών πρέπει να καθιδρύσουμε μια βέλτιστη στρατηγική για την τοποθέτηση των αντικειμένων στους υποκατάστατους εξυπηρετητές λαμβάνοντας υπόψη την χωρητικότητα C του κάθε εξυπηρετητή και το μέγεθος του κάθε αντικειμένου έτσι ώστε να ελαχιστοποιείται ο χρόνος απόκρισης, αυτό θα επιτευχθεί εάν οι υποκατάστατοι εξυπηρετητές σε μια περιοχή P βρίσκονται σε θέση να εξυπηρετήσουν ένα εξαιρετικά μεγάλο ποσοστό των αιτήσεων από τους κόμβους Q που βρίσκονται σε αυτή την περιοχή.

Έστω ότι δημιουργούμε μια συνάρτηση $Put(r_i, N, C_i \forall i \in N, Predict(G,P, O,R), S_i \forall i \in O)$ δίνει ένα σύνολο αντικειμένων που θα τοποθετηθούν στον κάθε υποκατάστατο εξυπηρετητή κάθε περιοχής r_i έτσι ώστε το $\frac{Q_{hit}}{Q_{total}}$ να μεγιστοποιείται και να ισχύει ο περιορισμός

$$\sum_{\forall i \in O} S_i f_{ik} \leq C_k$$

όπου $f_{ik} = \begin{cases} 1 & \text{εάν το αντικείμενο } i \text{ βρίσκεται αποθηκευμένο στον υποκατάστατο} \\ & \text{εξυπηρετητή } k \end{cases}$

0 εάν δεν βρίσκεται

Κεφάλαιο 7

Συμπεράσματα

7.1 Συμπεράσματα	46
7.2 Μελλοντική εργασία	46

7.1 Συμπεράσματα

Έχουμε μελετήσει και αναλύσει τη σχετική βιβλιογραφία εντοπίζοντας ότι τα μοντέλα που προταθήκαν μέχρι τώρα υπολείπονται και χρήζουν βελτίωσης.

Καθορίσαμε αυστηρά το πρόβλημα θέτοντας τους περιορισμούς στους οποίους πρέπει να υπακούει η λύση που θα δώσουμε.

Έχουμε συλλέξει και αναλύσει τα κοινωνικά δεδομένα που χρειαζόμαστε. Έχουμε βρει το πόσο επηρεάζονται οι χρήστες στα κοινωνικά δίκτυα. Αν πάρουμε απλά το συντελεστή διάδοσης φαίνεται ότι οι χρήστες δεν επηρεάζονται τόσο πολύ από το κοινωνικό δίκτυο, αλλά έχουμε βρει και εντοπίσει τους παράγοντες οι οποίοι αυξάνουν την επιρροή του κοινωνικού δικτύου. Ακόμα έχουμε βρει την καθυστέρηση διάδοσης και για πόσο χρόνο μπορεί να επηρεάσει μια δημοσίευση σε ένα κοινωνικό δίκτυο.

Έχουμε δώσει μία ολοκληρωμένη και τεκμηριωμένη λύση στο πρόβλημα του ελέγχου και δοκιμής ενός συστήματος που επιλύει το πρόβλημα το οποίο προσπαθούμε να επιλύσουμε. Η γεννήτρια αιτήσεων μοντελοποιεί τις αιτήσεις χρησιμοποιώντας πραγματικές κατανομές πραγματικά δεδομένα και ένα ρεαλιστικό μοντέλο διάδοσης.

7.2 Μελλοντική εργασία

Δείχνοντας ότι οι χρήστες επηρεάζονται από το κοινωνικό δίκτυο πρέπει να δημιουργήσουμε ένα σύστημα το οποίο να εκμεταλλεύεται αυτές τις πληροφορίες για να επιλύει το πρόβλημα της επιλογής των υποκατάστατων εξυπηρετητών που θα αντιγράφονται τα αντικείμενα.

Ακόμα χρειάζεται να δούμε ποιοι άλλοι παράγοντες επηρεάζουν την κοινωνική διάδοση και να βρούμε ένα τρόπο να υπολογίζεται η πιθανότητα να υιοθετήσει ένας χρήστης κάποιο περιεχόμενο ξεχωριστά κάθε φορά δίνοντας καλύτερα και πιο ακριβή αποτελέσματα.

Η γεννήτρια αιτήσεων πρέπει να τροποποιηθεί ούτως ώστε να μπορεί να λάβει ένα διαφορετικό συντελεστή διάδοσης για κάθε πιθανή διάδοση έτσι ώστε τα αποτελέσματα να είναι ακόμα πιο κοντά στην πραγματικότητα.

Τέλος πρέπει να ολοκληρωθεί η συλλογή των δεδομένων έτσι ώστε να επιβεβαιώσουμε ότι τα συμπεράσματα που βγάλαμε μέχρι τώρα είναι έγκυρα.

Βιβλιογραφία

- [1] George Pallis and Athena Vakali. 2006. Insight and perspectives for content delivery networks. *Commun. ACM* 49, 1 (January 2006), 101-106.
- [2] Meeyoung Cha, Alan Mislove, Ben Adams, and Krishna P. Gummadi. 2008. Characterizing social cascades in flickr. In *Proceedings of the first workshop on Online social networks (WOSN '08)*.
- [3] Nishanth Sastry, Eiko Yoneki, and Jon Crowcroft. 2009. Buzztraq: predicting geographical access patterns of social cascades using social networks. In *Proceedings of the Second ACM EuroSys Workshop on Social Network Systems (SNS '09)*.
- [4] Salvatore Scellato, Cecilia Mascolo, Mirco Musolesi, Jon Crowcroft. Track Globally, Deliver Locally: Improving Content Delivery Networks by Tracking Geographic Social Cascades. In *Proceedings of 20th International World Wide Web Conference (WWW 2011)*. Hyderabad, India. March 2011.
- [5] M. Cha, H. Kwak, P. Rodriguez, Y-Y. Ahn, and S. Moon. I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System. In *Proc. of ACM IMC*, 2007.
- [6] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and Analysis of Online Social Networks. In *Proc. of ACM IMC*, 2007.
- [7] Yu, H., Zheng, D., Zhao, B. Y., and Zheng, W. Understanding user behaviour in large-scale video-on-demand systems. *SIGOPS Oper. Syst. Rev.* 40, 4 (2006), 333-344.
- [8] Phillipa Gill, Martin Arlitt, Zongpeng Li, and Anirban Mahanti. Youtube Traffic Characterization: a View from the Edge. In *Proceedings of IMC '07*, pages 15-28, New York, NY, USA, 2007. ACM.
- [9] F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida. Characterizing User Behavior in Online Social Networks. In *IMC*, 2009.

- [10] Xu Cheng; Dale, C.; Jiangchuan Liu, "Statistics and Social Network of YouTube Videos," Quality of Service, 2008. IWQoS 2008. 16th International Workshop on , vol., no., pp.229-238, 2-4 June 2008
- [11] R. Torres, A. Finamore, J. Kim, M. Mellia, M. M. Munafo, and S. Rao. Dissecting video server selection strategies in the youtube cdn. ECE Technical Reports 408, Purdue University, January 2011.
- [12] S. Traverso, K. Huguein, I. Trestian, V. Erramilli, N. Laoutaris, K. Papagiannaki. TailGate: Handling Long-Tail Content with a Little Help from Friends WWW April 2012
- [13] MacQueen, J. B. Some methods for classification and analysis of multivariate observations. In Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability (1967).
- [14] Minas Gjoka, Maciej Kurant, Carter T. Butts, and Athina Markopoulou. A walk in Facebook: Uniform sampling of users in online social networks. In Proc. of the IEEE Infocom, March 2010.
- [15] M. Zink, K. Suh, Y. Gu, and J. Kurose, "Watch Global, Cache Local: YouTube Network Traffic at a Campus Network - Measurements and Implications," in Proc. of SPIE/ACM Multimedia Computing and Networking
- [16] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP Topologies with Rocketfuel. In Proceedings of the ACM SIGCOMM Conference, pages 133–146, Pittsburgh, PA, August 2002.
- [17] H. Kwak, C. Lee, H. Park, and S. Moon. 2010. What is Twitter, a social network or a news media? In Proceedings of WWW. Conference, April 26-30, 2010
- [18] <https://dev.twitter.com/>
- [19] <https://github.com/sixohsix/twitter>
- [20] <http://an.kaist.ac.kr/traces/WWW2010.html>
- [21] <http://unshort.me/api.html>

- [22] <http://unshort.tk/api.php>
- [23] https://developers.google.com/youtube/2.0/developers_guide_protocol#Retrieve_video_entry
- [24] <https://developers.google.com/maps/documentation/geocoding/>
- [25] Measuring a S. Triukose, Z. Wen, and M. Rabinovich. Measuring a Commercial Content Delivery Network. In Proc. WWW,2011
- [26] Eytan Bakshy, Itamar E. Bakshy, I. Rosenn, C. Marlow, and L. Adamic. The role of social networks in information diffusion. In WWW, 2012.
- [27] M. Cha, H. Kwak, P. Rodriguez, Y-Y. Ahn, and S. Moon. I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System. In Proc. of ACM IMC, 2007
- [28] R. Torres, A. Finamore, J. Kim, M. Mellia, M. M. Munafo, and S. Rao. Dissecting video server selection strategies in the youtube cdn. ECE Technical Reports 408, Purdue University, January 2011
- [29] Yu, H., Zheng, D., Zhao, B. Y., and Zheng, W. Understanding user behaviour in large-scale video-on-demand systems. SIGOPS Oper. Syst. Rev. 40, 4 (2006), 333-344.
- [30] P. Gill et al., "Characterizing User Sessions on YouTube," IEEE Multimedia Computing and Networking, 2008.
- [31] A. Abhari and M Soraya. Workload generation for YouTube. Multimedia Tools and Applications, 46(1):91-118,2010.

Παράρτημα Α

Εδώ παρουσιάζεται ο κώδικας του ιχνηλάτη ο οποίος ιχνηλατεί τον κοινωνικό γράφο.

```
import sys
import twitter
import time
import urllib2

from recipe__make_twitter_request import make_twitter_request
from recipe__oauth_login import oauth_login
import functools

#proxy = urllib2.ProxyHandler({'https': 'http://www.cs.ucy.ac.cy:8008'})
#opener = urllib2.build_opener(proxy)
#urllib2.install_opener(opener)

#t = twitter.Twitter(domain='api.twitter.com', api_version='1')
with open("startUser.txt", "r") as f1:
    start = int(f1.readline())
    print "Start User:",
    print start

Requests=0
CurrOAuth=0
#maxOAuth=48;
maxOAuth=10;
CurrRequests=0;
APP_NAME = 'giorkosmanApp'
CONSUMER_KEY = '1WZgi3Rs0wxZjKbslZdg'
CONSUMER_SECRET = 'q9r3cqJ3xKKhOJHZNOmWSkNWMlAKQdArKJEk9axGPk'
t = oauth_login(APP_NAME, CONSUMER_KEY, CONSUMER_SECRET,str(CurrOAuth))
xxiTCCounter=0;
TCflag=1;
while TCflag==1:
    try:
```

```

        CurrRequests=t.account.rate_limit_status()['remaining_hits']
        TCflag=0
    except:
        xxiTCCounter+=1
        if xxiTCCounter==5:
            raise
        print >> sys.stderr, 'Twitter1 error: sleeping for %i secs' % (10)
        time.sleep(10)
        continue;

with open("../users.txt", "r") as f:
    line = f.readline()
    while line != "": ##While there are friends
        # Do stuff with line.
        currUser=int(line)
        if currUser<start:
            line = f.readline()
            continue
        getFollowerIds = functools.partial(make_twitter_request, t, t.followers.ids)
        cursor = -1
        ids = []
        while cursor != 0:
            flag=1
            #Use make_twitter_request via the partially bound callable...
            if CurrRequests == 50 or CurrRequests == 150 or CurrRequests==101 or
CurrRequests==202 or CurrRequests<13:
                CurrOauth=(CurrOauth+1)%maxOauth
                t = oauth_login(APP_NAME, CONSUMER_KEY,
CONSUMER_SECRET,str(CurrOauth))
                xxiTCCounter=0;
                TCflag=1;
                while TCflag==1:
                    try:

```

```

CurrRequests=t.account.rate_limit_status()['remaining_hits']
        TCflag=0
    except:
        xxiTCCounter+=1
        if xxiTCCounter==5:
            raise
        print >> sys.stderr, 'Twitter2 error: sleeping for %i secs'
% (10)

        time.sleep(10)
        continue;

    getFollowerIds = functools.partial(make_twitter_request, t, t.followers.ids)
    print >> sys.stderr, 'Switch to: %i.' % (CurrOauth)
xxiTCCounter=0;
    TCflag=1;
    while TCflag==1:
        try:
            response = getFollowerIds(user_id=currUser, cursor=cursor)
            TCflag=0
        except:
            xxiTCCounter+=1
            if xxiTCCounter==5:
                raise
            print >> sys.stderr, 'Twitter3 error: sleeping for %i secs' % (10)
            time.sleep(10)
            continue;

    CurrRequests-=1;
    Requests+=1
    try:
        ids += response['ids']
        cursor = response['next_cursor']

        print >> sys.stderr, 'Fetched %i total ids for %s' % (len(ids), str(currUser))
    except Exception:

```

```

    flag=0
    cursor=0
    continue

#print "Requests so far:",
#print str(Requests)
# Consider storing the ids to disk during each iteration to provide an
# an additional layer of protection from exceptional circumstances

# Do something useful with the ids like store them to disk...
if flag==1:
    with open("Graph.txt", "a") as fG:
        fG.write(str(currUser))
        fG.write("\n")
        fG.write(str(len(ids)))
        fG.write("\n")
        for i in range(len(ids)):
            fG.write(str(ids[i]))
            fG.write("\n")
            fG.write("!!!!!!!!!!!!\n")
else:
    with open("SkippedUsers.txt", "a") as fS:
        fS.write(str(currUser))
        fS.write("\n")
with open("startUser.txt", "w") as f2:
    f2.write(str(currUser+1))
    f2.write("\n")
line = f.readline()

```

Παράρτημα Β

Εδώ παρουσιάζεται ο κώδικας της κύριας μεθόδου της γεννήτριας αιτήσεων

```
#include <cstdlib>
#include <cstdio>
#include<math.h>
#include<time.h>

#include<random>
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <list>
#include <map>

#include"GetDay.h"
#include "Distributions.h"
#include "Videos.h"
#include "Session.h"
#include "User.h"

using namespace std;

#define HOURSOFSIMULATION 168;
//Per Region
int main(int argc, char *argv[]){
    srand((unsigned)time(NULL));

    Distributions distributions;
    Videos videos;
    long userCount;
    User* allUsers;

    int UTC_Offset;
    //cin>>UTC_Offset;
    //UTC_Offset=0;
    allUsers=User::readAllFile(userCount,UTC_Offset,argv[1]);

    //cout<<userCount<<endl;

    vector<vector<Session*>*> activeSessions(1000000);
    for(int i=0;i<1000000;++i){
        activeSessions[i]=new vector<Session *>;
    }

    //5/3/2012;

    int day;
    if(UTC_Offset<0){
        day=GetDate(1,3,2012);
    }
    else{
        day=GetDate(2,3,2012);
    }
    //cout<<day<<endl;
```

```

int time=0;
long timestamp=0;
int hoursToBeRUN=HOURSOFSIMULATION;
long sessionsForDay;
Session *daySessions;
int seconds=0;
int numDays=hoursToBeRUN/24;
int currentSessionCounter=0;

int StartHour;
if(UTC_Offset<0){
    StartHour=24+(UTC_Offset/3600);
}
else{
    StartHour=0+(UTC_Offset/3600);
}

//cout<<StartHour<<endl;
bool first=true;

int theTimeOfDay=StartHour;
ifstream inFile;
inFile.open ("Files/10millionRandomNumbers.txt");
string line;

bool flagFirstDay=true;
if(StartHour!=0){
    flagFirstDay=false;
    sessionsForDay=distributions.GetWeeklyRequests(day)*userCount;
    daySessions=new Session[sessionsForDay];
    for(int i=0;i<sessionsForDay;++i){
        getline(inFile,line);

        daySessions[i].set(atof(line.c_str())*userCount,distributions.getSessionDuration
n());
    }
}
ofstream myfile;
myfile.open (argv[2]);

cout<<"Day="<<day<<endl;
cout<<"H="<<theTimeOfDay<<endl;
for(int h=0;h<hoursToBeRUN;++h,theTimeOfDay=(theTimeOfDay+1)%24){
    cout<<"H="<<theTimeOfDay<<endl;
    if(theTimeOfDay==0){
        if(flagFirstDay==false){
            day=(day+1)%7;
            cout<<"Day="<<day<<endl;
        }
        flagFirstDay=false;
        sessionsForDay=distributions.GetWeeklyRequests(day)*userCount;
        daySessions=new Session[sessionsForDay];
        for(int i=0;i<sessionsForDay;++i){
            getline(inFile,line);
            //cout<<line.c_str()<<"-"<<atof(line.c_str())<<"-
"<<atof(line.c_str())*userCount<<endl;

            daySessions[i].set(atof(line.c_str())*userCount,distributions.getSessionDuration
n());
        }
        currentSessionCounter=0;
    }
}

```

```

        int sessionsPerHour=
sessionsForDay*distributions.GetHourlyRequests(theTimeOfDay);
        for(int i=0;i<sessionsPerHour;++i){
            if(currentSessionCounter>=sessionsForDay)
                break;
            getline(inFile,line);
            activeSessions[atoi(line.c_str())*3600]-
>push_back(&daySessions[currentSessionCounter]);
            ++currentSessionCounter;
        }

        Session *tempSession;
        int currUser;
        int *vid;
        //int miniCounter=0;
        for(int i=0;i<3600;++i,++timestamp){
            for(int j=0;j<activeSessions[i]->size();++j){
                tempSession=activeSessions[i]->at(j);
                currUser=tempSession->getUser();
                getline(inFile,line);
                //cout<<"L:"<<line;
                //cout<<"--X:"<<currUser<<endl;
                //User *x=&allUsers[currUser];

                vid=videos.selectVideo(allUsers[currUser].getSeenVideos(timestamp),atoi(line.c_
str()));

                tempSession->decreaseDuration(vid[1]);
                if(tempSession->canSeeMoreVideos())
                    activeSessions[i+vid[1]]->push_back(tempSession);

                //cout<<"Timestamp"<<timestamp<<"User"<<allUsers[currUser].getID()<<"Video"<<vi
d[0]<<endl;

                myfile <<
timestamp<<"\t"<<allUsers[currUser].getID()<<"\t"<<vid[0]<<endl;
                //miniCounter++;
            }
            activeSessions[i]->clear();
        }
//        cout<<miniCounter;
        for(int i=0;i<1000000;++i){
            if(i+3600<1000000)
            {
                //delete(activeSessions[i]);
                activeSessions[i]=activeSessions[i+3600];
                //activeSessions[i]->clear();
                //activeSessions[i]->insert( activeSessions[i]->end(),
activeSessions[i+3600]->begin(), activeSessions[i+3600]->end() );
            }
            else
                activeSessions[i]=new vector<Session *>;
        }

        myfile.close();
        inFile.close();
        //cout<<"Fin";
        //char wait;
        //scanf("%c",&wait);
        return 0;
}

```

Παράρτημα Γ

Εδώ παρουσιάζεται ο κώδικας ο οποίος υπολογίζει τον συντελεστή διάδοσης.

```
import java.io.BufferedReader;
import java.io.DataInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.InputStreamReader;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.HashMap;
import java.util.Locale;
import java.util.StringTokenizer;
import java.util.TimeZone;

public class GetCascadeCoefficientOnly {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        //String x="Thu Oct 14 15:18:42 +0000 2010";
        System.out.println("Running Get Cascade Coefficient only");
        SimpleDateFormat dateFormat = new SimpleDateFormat("EEE MMM dd
HH:mm:ss ZZZZZ yyyy",Locale.US);
        dateFormat.setTimeZone(TimeZone.getTimeZone("Etc/UTC"));
        Date theOldestAllowedDate2=null;
        try {
```

```

        theOldestAllowedDate2=dateFormat.parse("Tue Mar 21 12:50:00
+0000 2006");
    } catch (ParseException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
    Date theOldestAllowedDate=null;
    try {
        theOldestAllowedDate=dateFormat.parse("Sun Jan 1 00:00:00 +0000
2012");
    } catch (ParseException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }

    Date lim=null;
    try {
        lim=dateFormat.parse("Fri Mar 02 00:00:00 +0000 2012" );
    } catch (ParseException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }

```

```

        Calendar                                aCalendar                                =
Calendar.getInstance(TimeZone.getTimeZone("Etc/UTC"));
        aCalendar.setTime(lim);
        //aCalendar.add(Calendar.DATE, 0);
        //lim=aCalendar.getTime();

        aCalendar.add(Calendar.DATE,7);
        Date downLim1=aCalendar.getTime();
        aCalendar.add(Calendar.DATE, 3);
        Date downLim2=aCalendar.getTime();

```

```

System.out.println(lim);
System.out.println(downLim1);
System.out.println(downLim2);

//System.exit(0);

// "Fri Mar 02 14:50:00 EET 2012"
int CascadedUserCount=0;
int activeCascadedUserCount=0;
double theTotalCascadeCoefficient=0;
ArrayList<CascadedItem> inItems=new ArrayList<CascadedItem>();
ArrayList<CascadedItem> outItems=new ArrayList<CascadedItem>();
HashMap<String, CascadedItem> onlyOnce=new HashMap<String,
CascadedItem>();
CascadedItem tempCascadeItem;
CascadedItem veryTempCascadeItem;
for (int k = 0; k < args.length-1; k++) {
    System.out.println("Cascaded Users:"+CascadedUserCount);
    System.out.println("Active Cascaded
Users:"+activeCascadedUserCount);
    System.out.println("Total cascade
coefficient:"+theTotalCascadeCoefficient);
    System.out.println("The per user cascade
coefficient:"+theTotalCascadeCoefficient/activeCascadedUserCount);

    String path = args[k];

    String fileName;
    File folder = new File(path);
    File[] listOfFiles = folder.listFiles();

    int newCompletion=0;
    int Completion=0;

```

```

        System.out.println("Current Folder is:"+ path);
        for (int i = 0; i < listOfFiles.length; i++)
        {
            newCompletion=(int)
Math.round(((double)i*100/listOfFiles.length);
            if(newCompletion!= Completion){
                System.out.println("Progress:"+newCompletion+"%");
                Completion=newCompletion;
            }
            if (listOfFiles[i].isFile())
            {

                fileName = listOfFiles[i].getName();

                File tF=new File(args[args.length-1]+"/"+fileName);
                if (!tF.exists())
                    continue;

                inItems.clear();
                outItems.clear();

                onlyOnce.clear();
                try{

                    FileInputStream      fstream      =      new
FileInputStream(path+"/"+fileName);

                    // Get the object of DataInputStream
                    DataInputStream      in            =      new
DataInputStream(fstream);

                    BufferedReader br = new BufferedReader(new
InputStreamReader(in));

                    String strLine;
                    //Read File Line By Line

```

```

String vid,s1;
while ((strLine = br.readLine()) != null) {
    // Print the content on the console
    if(strLine.equals(""))
        continue;
}
StringTokenizer tokkk=new
StringTokenizer(strLine,"\t\n");

try{
    s1=tokkk.nextToken();
    tokkk.nextToken();
    tokkk.nextToken();
    vid=tokkk.nextToken();
} catch (Exception e) {

    continue;
}
Date d2=null;
//String x="Thu Oct 14 15:18:42 +0000
2010";

try {

    d2=dateFormat.parse(s1);
    if (d2.before(lim)
        d2.after(downLim2))
        continue;
    //if
    (d2.before(theOldestAllowedDate))
        continue;

    //System.out.println(dateFormat.format(d2));

    //System.out.println(d2.toString());

```

block

```
        } catch (ParseException e) {
            // TODO Auto-generated catch

            e.printStackTrace();

            System.err.println("Line:"+strLine);

            continue;

        }

        if(vid.contains("#"))

            vid=vid.substring(0,vid.indexOf("#"));

        if(vid.contains("."))

            vid=vid.substring(0,vid.indexOf("."));

        if(vid.contains("\\"))

            vid=vid.substring(0,vid.indexOf("\\"));

        if(vid.contains("\'"))

            vid=vid.substring(0,vid.indexOf("\'"));

        if(vid.contains(":"))

            vid=vid.substring(0,vid.indexOf(":"));

        if(vid.length()>12 || vid.length()<6)

            continue;
```

```

tempCascadeItem=onlyOnce.get(vid);
if(tempCascadeItem==null){
    tempCascadeItem= new
    CascadedItem(vid, d2);

    outItems.add(tempCascadeItem);
    onlyOnce.put(vid,
tempCascadeItem);
}
else{
    veryTempCascadeItem =new
    CascadedItem(vid, d2);

    if(veryTempCascadeItem.compareTo(tempCascadeItem)==-1){

//System.out.print("Out:"+tempCascadeItem);

tempCascadeItem.reSetDate(veryTempCascadeItem.date);
//System.out.println("-
>"+tempCascadeItem);

    }
    //if(tempCascadeItem.)
}

}
//System.out.println(vid);

br.close();
in.close();
fstream.close();
} catch (Exception e){//Catch exception if any
    System.err.println("Error: " +
e.getMessage()+"!\n"+e.toString()+"!\n");
    e.printStackTrace();
    System.exit(-2);

```

```

    }
    onlyOnce.clear();
    try{
        FileInputStream fstream = new
FileInputStream(args[args.length-1]+"/"+fileName);
        // Get the object of DataInputStream
        DataInputStream in = new
DataInputStream(fstream);
        BufferedReader br = new BufferedReader(new
InputStreamReader(in));

        String strLine;
        //Read File Line By Line
        String vid,s1;
        while ((strLine = br.readLine()) != null) {
            // Print the content on the console
            if(strLine.equals("")) ||
strLine.isEmpty()){
                continue;
            }
            StringTokenizer tokkk=new
StringTokenizer(strLine,"\t\n");

            try{
                s1=tokkk.nextToken();
                tokkk.nextToken();
                tokkk.nextToken();
                vid=tokkk.nextToken();
            }catch (Exception e) {
                continue;
            }
            Date d2=null;
            //String x="Thu Oct 14 15:18:42 +0000
2010";

            try {

```

```

d2=dateFormat.parse(s1);
//if
(d2.before(theOldestAllowedDate))

//      continue;
if      (d2.before(lim)      ||

d2.after(downLim1))

      continue;

//System.out.println(dateFormat.format(d2));

//System.out.println(d2.toString());

} catch (ParseException e) {
    // TODO Auto-generated catch
block

    e.printStackTrace();

System.err.println("Line:"+strLine);

    continue;
}

if(vid.contains("#"))

vid=vid.substring(0,vid.indexOf("#"));

if(vid.contains("."))

vid=vid.substring(0,vid.indexOf("."));

if(vid.contains("\\"))

vid=vid.substring(0,vid.indexOf("\\"));

if(vid.contains("\\"))

```

```

vid=vid.substring(0,vid.indexOf("\\"));

if(vid.contains(":"))

vid=vid.substring(0,vid.indexOf(":"));


if(vid.length()>12 || vid.length()<6)
    continue;

tempCascadeItem=onlyOnce.get(vid);
if(tempCascadeItem==null){
    tempCascadeItem=        new
    CascadedItem(vid, d2);

    inItems.add(tempCascadeItem);
    onlyOnce.put(vid,
tempCascadeItem);
}
else{
    veryTempCascadeItem    =new
    CascadedItem(vid, d2);

    if(veryTempCascadeItem.compareTo(tempCascadeItem)==1){

//System.out.print("In:"+tempCascadeItem);

tempCascadeItem.reSetDate(veryTempCascadeItem.date);

//System.out.println("-
>"+tempCascadeItem);

    }
    //if(tempCascadeItem.)
}

```

```

    }
    //System.out.println(vid);

    br.close();
    in.close();
    fstream.close();
} catch (Exception e){//Catch exception if any
    System.err.println("Error: " +
e.getMessage()+"!\n"+e.toString()+"!\n");
    e.printStackTrace();
    System.exit(-2);
}
CascadedUserCount++;
//int count=0;

int currCascCount=0;
boolean flag=true;
int countedItems=0;
for (CascadedItem ci : inItems) {
    flag=true;
    for (CascadedItem co : outItems) {
        if(ci.compareTo(co)==1){
            //System.out.println("Set:"+ci+"-
>"+co);

            //++count;
            currCascCount++;
            countedItems++;
            flag=false;
            break;

        }
        else if(ci.vid.equals(co.vid)){
            flag=false;

```

```

        }

        /*
        else if(ci.compareTo(co)==-1){

//System.out.println("Rev:"+ci+"<-"+co);

                //++count;
        }
        */

    }
    //System.out.println(""+ci);
    if (flag){
        countedItems++;
    }

}

if(inItems.size()>0 && countedItems>0){

theTotalCascadeCoefficient+=(double)currCascCount/countedItems;

        activeCascadedUserCount++;
    }
    /*
    if(count>0){
        System.out.println("In size:"+inItems.size());
        System.out.println("Out size:"+outItems.size());
        try {
            Thread.sleep(10000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
    */
}

}

```

```
    }  
    System.out.println("Cascaded Users:"+CascadedUserCount);  
    System.out.println("Active Cascaded Users:"+activeCascadedUserCount);  
    System.out.println("Total cascade coefficient:"+theTotalCascadeCoefficient);  
    System.out.println("The          per          user          cascade  
coefficient:"+theTotalCascadeCoefficient/activeCascadedUserCount);  
  
    }  
  
}
```