



# International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

[www.iasir.net](http://www.iasir.net)

## Social Web Mining for Addressing Flash Crowds in CDNs

Aman Jhunjhunwala

Department of Computer Science,  
Birla Institute of Technology,  
Mesra, Jharkhand 835215, INDIA.

**Abstract:** The paper presents a survey of the current replica placement strategies on surrogates in Content Delivery Networks and proposes novel ways in which mining social networks can help derive useful insights in predicting user traffic on websites thereby helping CDNs to prepare for the abnormal spurts and traffic outages (called Shashdot Effect or Flash crowds). In this paper, we present metrics to extract vital information from tweets that we later feed to Machine Learning algorithms which eventually help in taking important decisions on moving user content around the CDN - composed of hundreds of surrogate servers scattered across the globe.

**Keywords:** Slashdot Effect, Content Delivery Networks, Social Web Mining, Data Mining, Machine Learning, Web Engineering

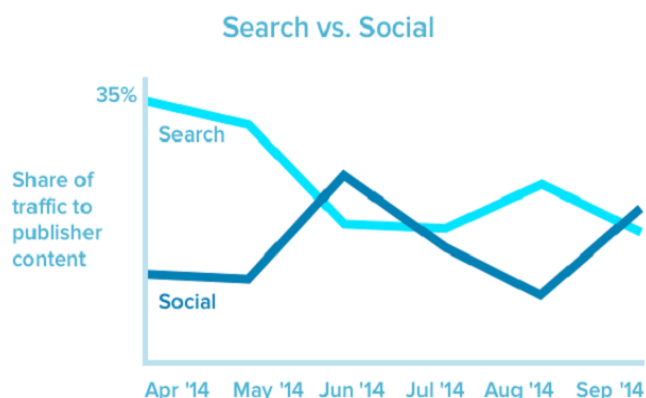
### I. Introduction

Content Delivery Networks (CDNs) [1]-[3] furnish services that enhance network performance by maximizing bandwidth, boosting accessibility and maintaining consistency through redundancy or content duplication. They offer reliable and swift applications and services by distributing content to multiple servers (called surrogate servers). Each CDN provider operates a globally distributed network of proxy storage servers deployed in multiple data centers – mostly places causing high traffic. CDNs form the backbone of a huge portion of the Web content today, including web objects (text, graphics and scripts), downloadable objects (media files, software, documents), applications (e-commerce, portals), live streaming media and on-demand streaming media. Besides better performance and availability, CDNs also offload the traffic served directly from the user's infrastructure to their own, resulting in tremendous cost savings for the user. CDNs provide a great degree of security and protection from sudden spurts and malicious DDoS attacks by using their large distributed server infrastructure to absorb the attack traffic.

Flash crowds or Slashdot effect, takes place when a website with massive popularity links to a smaller site, causing a massive increase in traffic. This overloads the smaller site, causing it to slow down or even temporarily become unavailable. The name stems from the huge influx of web traffic that would result from the technology news site Slashdot linking to websites. However, the name is somewhat dated as flash crowds from Slashdot were reported to be diminishing as of 2005 due to competition from similar sites. The effect has been associated with other websites or metablogs such as Facebook, Digg, Imgur, Reddit, and Twitter. Typically, less robust sites are unable to cope with the huge increase in traffic and become unavailable – common causes are lack of sufficient data bandwidth, servers that fail to cope with the high number of requests, and traffic quotas.

With the proliferation of social networks, a majority of website traffic today is driven by social media referrals.

Figure 1 Shareholio Data Comparing Traffic Referrals from Social Networks and Search



Personalities of repute and eminence, like Snowden have been responsible for bringing down entire Content Delivery Networks through just one tweet. Due to the influence celebrities and popular press exert on the masses, and the rapid rate at which the information propagates the social web, active analysis of major social mediums could help us predict, in advance, abnormal traffic spurts, and adopt means to mitigate the load on the server and provider network. Maintaining data on unnecessary duplicate servers is expensive, by analysing social content we can reduce the total costs incurred by the user while adapting to different traffic loads.

## **II. Survey of Existing CDN Mechanisms for Content Selection and Placement**

Since the location of replicas on surrogates is closely related to the content delivery process, the issue of choosing the best location for each surrogate is vital. The goal of optimal surrogate placement is to reduce user perceived latency for accessing content and to minimize the overall network bandwidth consumption for transferring replicated content from servers to clients. Current CDNs use the following approaches to Content Selection and Placement :-

### **A. K-Hierarchially Separated Tree (k-HST)**

This approach [9][10] models the server placement problem as the center placement problem which is defined as follows: for the placement of a given number of centers, minimize the maximum distance between a node and the nearest center. k-HST algorithm solves the server placement problem according to graph theory. In this approach, the network is represented as a graph  $G(V,E)$ , where  $V$  is the set of nodes and  $E \subseteq V \times V$  is the set of links. The algorithm consists of two phases. In the first phase, a node is arbitrarily selected from the complete graph (parent partition) and all the nodes which are within a random radius from this node form a new partition (child partition). The radius of the child partition is a factor of  $k$  smaller than the diameter of the parent partition. This process continues until each of the nodes is in a partition of its own. Thus the graph is recursively partitioned and a tree of partitions is obtained with the root node being the entire network and the leaf nodes being individual nodes in the network. In the second phase, a virtual node is assigned to each of the partitions at each level. Each virtual node in a parent partition becomes the parent of the virtual nodes in the child partitions and together the virtual nodes form a tree. Afterwards, a greedy strategy is applied to find the number of centers needed for the resulted k-HST tree when the maximum center-node distance is bounded by  $D$ .

### **B. Greedy Method**

The greedy algorithm [8] chooses  $M$  servers for duplication of content among  $N$  potential sites. In first iteration, the cost associated with each site is computed in the first iteration. It is assumed that access from all clients converges to the site under consideration. Hence, the lowest cost site is chosen. In the second iteration, the greedy algorithm searches for a second site (yielding the next lowest cost) in conjunction with the site already chosen. It requires the knowledge of the clients' locations in the network and all pair wise inter-node distances.

### **C. Topology Informed Placement Strategy**

In this placement strategy [7] servers are placed on candidate hosts in descending order of outdegrees (i.e. the number of other nodes connected to a node). Here the assumption is that nodes with more outdegrees can reach more nodes with smaller latency.

### **D. Tree Based Replica Placement**

The tree based algorithm [5] is based on the assumption that underlying topologies are trees. This algorithm models the replica placement as a dynamic programming model. In this approach, a tree is divided into several smaller trees  $T$  and the placement of  $k$  duplicates is achieved by placing  $t$  duplicates in the best way in each small tree  $T$ .

### **E. Hot Spot Replica Placement**

The hotspot algorithm [4] places replicas near the clients generating greatest load. It sorts the  $N$  potential sites according to the amount of traffic generated surrounding them and places replicas at the top  $M$  sites that generate maximum traffic.

### **F. Scalable Replica Placement**

This algorithm [6] generates replicas on demand and organizes them into an application-level multicast tree. This approach minimizes the number of replicas while meeting clients' latency constraints and servers' capacity constraints.

## **III. Proposed Methodology**

Throughout the research, we make persistent use of Metadata - content description that allows identification, discovery, and management of multimedia data. Metadata may also include information about the web sites that are currently accessing the file stored in the CDN. The following underline the major steps in building a content placement strategy on social web mining :-

### **A. Online Tweet Analysis**

As we have earlier described, analysing data from social platforms could help us derive key insights into Flash Crowds. For starters, a lot of people retweeting the link to the same website, signifies increasing popularity and hence increasing traffic. In this context, the number of tweets itself, or the "likes" or "retweets" of a tweet containing a link to the website can function as efficient metric for traffic determination. For this research, we

use Twitter's API and develop Python scripts to analyse the tweets in real-time. Given Twitter's ability to satisfy some fundamental human desires relating to communication and curiosity, some well-defined self-organizing behaviour emerges from its chaotic network dynamics.

Using the API's JSON object responses, we perform frequency analysis of the tweets, keeping track of 3 parameters – the URL being referred to by the tweet, the location of the user and keywords extracted from tweets.

For automated unsupervised keyword extraction from tweets, Marujo et al. propose a state of the art effective method[11] for extracting keywords from tweets. Although the baseline TF-IDF algorithms do not work well for short tweets where most terms appear only once. The MAUI toolkit- indexer[12] extension proposes the use of Brown Clustering, Continuous Word Vectors and Keyword Prediction for better extraction.

Brown clusters [13] are effective for clustering lexical variants. Words such as “yes”, “yep” and “yess” are generally inserted into the same cluster as they occur in similar contexts. It also builds a hierarchical structure for clusters.

Continuous word vectors [14][15] learn word representations from neural language models in which a hidden layer is defined that maps words into a continuous vector. In these models, a hidden layer is defined that maps words into a continuous vector. The parameters of this hidden layer are estimated by maximizing a goal function, such as the likelihood of each word predicting surrounding words. They use a structured skip-gram goal function [15] and for each word they extracted its respective word features.

A last problem is the high variance in terms of the number of keywords per tweet. An extractor extracting all words as keywords would be incorrect. A better approach would be to learn a model to predict the number of keywords using a training set based on a set of features. This is done using linear regression, which extracts a feature set from an input tweet  $f_1, \dots, f_n$  and returns the expected number of keywords in the tweet. As features we selected the number of words in the input tweet with the intuition that the number of keywords tends to depend on the size of the tweet.

This extraction process has an F1 measure of 76.81 (P=71.81, R=82.55) on a Test set and 73.62 (P=67.55, R=82.55) on Dev Set.

#### **B. Predicting Surrogates for Replica Placement**

Using semantic similarity metrics we can correlate trending words with words in the metadata for CDN files. For example, if some major breakthrough takes place during the UN Environmental Conference, it will be natural for people to visit news site covering the full story. Traffic at those particular pages will drastically rise. Those files that have metadata keywords similar to those trending on social media (like “environment”, “UN conference”, etc in the above case) will be replicated on surrogates that belong to the region where the topics are trending. Apart from user provided metadata, webpages that call a CDN resource often, can be crawled and custom keywords are generated by scraping the contents of the webpage. This way the metadata is always fresh and new articles can be easily added without requiring manual updation of site metadata.

The aforesaid methodology addresses flash crowd problems by making the requisite number of copies before traffic spurts. The existing dynamic algorithms kick in only after its quite late and the network is already congested.

Additionally, as an aside, files can also be clustered into groups of similar items using Clustering algorithms (k-Means, SVMs, etc) or Basket Analysis (through association rules). Whenever a large number of files from the cluster are being replicated due to high demand, other files that are present in the cluster are also duplicated on the same surrogates. Belonging to the same group and dealing with similar topics, their demand is expected to rise in the near future as well, and we can prepare in advance for the upsurge.

#### **C. Helping better placement of Cold Items**

Cold CDN items are items that are fresh to the CDN – files that have just have been just been added. By applying k-Nearest Neighbours algorithm with the metadata keywords as features, we find files that closely resemble the cold file (having common or similar metadata). We then look at the list of surrogates where each of the similar files are currently duplicated on. We store the replicas of the cold file on the surrogates that are most common on the list (exceeding a particular threshold). This method helps to anticipate the traffic that might arrive for the cold item more effectively and efficiently.

### **IV. Conclusion**

A high degree of correlation has been observed in social media posts and traffic on websites. By integrating the proposed techniques, not only can CDN providers offload the manual supervision to a certain extent but also collect valuable data that can help them and other researchers better understand online user behavior and eventually predict Internet traffic successfully – which will be of tremendous value to cloud companies and users who can provision extra processing units in anticipation of higher loads. Further research in this area to capture the finer intricacies of the relationship is required. As the social web grows, it can be scanned to derive useful insights in unexpected fields and areas. With the next generation CDNs, often called Content Service Networks rapidly being prototyped and matured, this research presents vital components that can be used to better handle

traffic anomalies. In the nascent but rapidly evolving domain of Peer-Networked Content Distribution implementation of Machine Learning Techniques for advanced load balancing could lead to significant performance improvements for the peers. This research paper begins an embryonic journey into the applications of Machine Learning in Internet Management, one that holds tremendous prospect and promise into the future.

### References

- [1] G. Peng, "CDN: Content Distribution Network," Technical Report TR-125, Experimental Computer Systems Lab, Department of Computer Science, State University of New York, Stony Brook, NY, 2003.<http://citeseer.ist.psu.edu/peng03cdn.html>
- [2] A. Vakali, and G. Pallis, "Content Delivery Networks: Status and Trends," IEEE Internet Computing, IEEE Computer Society, pp. 68-74, November-December 2003
- [3] G. Pallis, and A. Vakali, "Insight and Perspectives for Content Delivery Networks," *Communications of the ACM*, Vol. 49, No. 1, ACM Press, NY, USA, pp. 101-106, January 2006.
- [4] L. Qiu, V. N. Padmanabhan, and G. M. Voelker, "On the Placement of Web Server Replicas," In *Proceedings of IEEE INFOCOM*, Anchorage, Alaska, USA, pp. 1587-1596, April 2001
- [5] B. Li, M. J. Golin, G. F. Italiano, D. Xin, and K. Sohrawy, "On the Optimal Placement of Web Proxies in the Internet," In *Proceedings of IEEE INFOCOM*, NY, USA, pp. 1282-1290, March 1999.
- [6] Y. Chen, R. H. Katz, and J. D. Kubiatowicz, "Dynamic Replica Placement for Scalable Content Delivery," In *Proceedings of International Workshop on Peer-to-Peer Systems (IPTPS 02)*, LNCS 2429, Springer-Verlag, pp. 306-318, 2002.
- [7] S. Jamin, C. Jin, A. R. Kure, D. Raz, and Y. Shavitt, "Constrained Mirror Placement on the Internet," In *Proceedings of IEEE INFOCOM*, Anchorage, Alaska, USA, April 2001.
- [8] P. Krishnan, D. Raz, and Y. Shavitt, "The Cache Location Problem," *IEEE/ACM Transaction on Networking*, Vol. 8, No. 5, 2000
- [9] S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, "On the placement of Internet Instrumentation," In *Proceedings of IEEE INFOCOM*, Tel-Aviv, Israel, pp. 295-304, March 2000.
- [10] Y. Bartal, "Probabilistic Approximation of Metric Space and its Algorithmic Applications," In *Proceedings of 37<sup>th</sup> Annual IEEE Symposium on Foundations of Computer Science*, October 1996
- [11] L Marujo, W Ling, I Trancoso, C Dyer, A Black, A Gershman, D Martins de Matos, J P. Neto and J Carbonell, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers)*, pages 637–643, Beijing, China, July 26-31, 2015.
- [12] Olena Medelyan, Vye Perrone, and Ian H. Witten, 2010, Subject metadata support powered by maui. In Jane Hunter, Carl Lagoze, C. Lee Giles, and Yuan-Fang Li, editors, JCDL, pages 407–408. ACM
- [13] Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *HLT-NAACL*, pages 380–390.
- [14] Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12
- [15] Wang Ling, Chris Dyer, Alan Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.