

Technical Disclosure Commons

Defensive Publications Series

November 28, 2018

Method Of Caching Ad Requests In Ad Serving Systems

Tuna Toksoz

William Coster

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation

Toksoz, Tuna and Coster, William, "Method Of Caching Ad Requests In Ad Serving Systems", Technical Disclosure Commons, (November 28, 2018)

https://www.tdcommons.org/dpubs_series/1711



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

METHOD OF CACHING AD REQUESTS IN AD SERVING SYSTEMS

ABSTRACT

A system and method are disclosed, for generating a pseudo-auction for caching ad requests based on network costs. The method includes analyzing the apps used by the user along with the tuples of associated app metadata, ad targeting criteria, ads served, etc. The ads are downloaded and stored in a temporary cache. When it is time to show an ad, the cache is queried for available ads. If the ad serving system indicates availability of any of the creatives picked up, these are retrieved from cache without hitting the network and displayed in the device. The method is implemented on an advertisement serving system connected to a mobile network serving multiple users. Advantages of the method include reduced latency and significant bandwidth saving.

BACKGROUND

Currently online advertising, and in particular advertisements received in mobile devices, are generated real time. To serve an ad, the request to get the ad and its assets is made at that moment in time. There are certain disadvantages associated with the ad serving method. First, since the request is made at that moment in time, there is a latency introduced in getting the required assets. A 1% increase in latency could result in as much as 1% revenue loss. Secondly, the user is likely to be connected to the internet through either his device's network or through Wi-Fi when the request is made. In case the device's network connection is used, the downloaded assets may cost money for the user.

DESCRIPTION

A system and method are disclosed, for generating a pseudo-auction for caching ad requests based on network costs. The method may be implemented on an advertisement

serving system connected to a mobile network serving multiple users. The system analyzes the user's activity log to identify the top ad creatives served to the user and may download and store the assets for the creatives in a temporary cache at a predetermined time when low cost bandwidth is available. When it is time to display an ad the cache is looked up first for available resources before hitting the network, thus reducing latency.

The method, as shown in FIG. 1 includes analyzing the user's activity log or other data sources that contain data that include ad request patterns, the apps that are used, the size of the app etc., in step A. In step B the system picks up at least the top 5 apps used by the user along with the tuples that include data on the size of the app, advertiser, etc. In step C, the top ad creatives already served are picked up. Assuming that the ad creative assets that were picked are likely to be served to the user, in step D the assets are downloaded and stored in a temporary cache. The asset is downloaded at a predetermined time, or based on a trigger such as when the network connectivity is switched from the phone network to Wi-Fi. In step E, when it is time to show an ad, the cache is queried for the available ads, and the ads are sent as a cue to the ad serving system. In step F, if the ad serving system returns any of the creatives picked up in step C, then in step G the cache is looked up for available resources. The ad is then displayed in the device in step H. If in step F, the ad serving system does not return any of the creatives then in step I, the asset is downloaded from the network and the ad is displayed.

In a variation of the method, instead of picking top creatives, the system may also run a simulated auction based on the tuple, or train a machine learning model to pick up more likely ads. The method may also be improved by using "cached ads" as another feature, and using the difference between actually served ad and cacheable ads at a given time.

Advantages of the disclosed method are reduced latency and significant bandwidth savings.

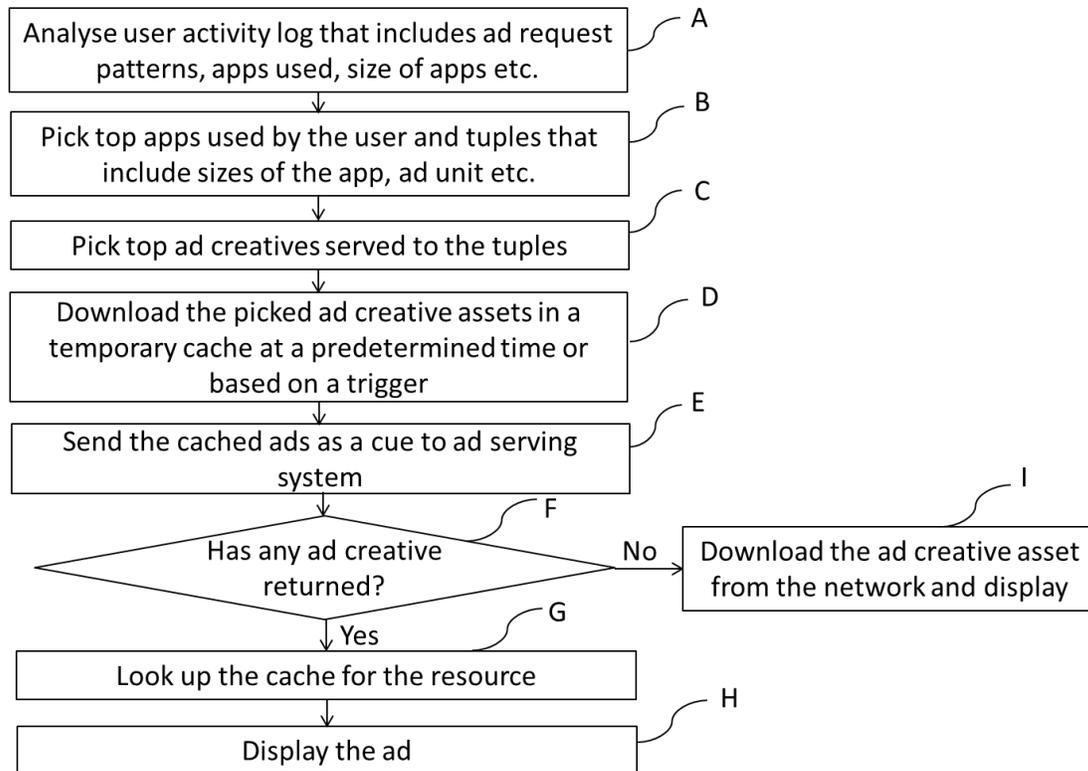


FIG. 1: Method of caching ad requests based on bandwidth availability