ANALYSIS OF USER IDENTIFICATION PROBLEMS IN RTB SYSTEMS

Ph.D. B.F. Trofimov[1], Dr. Sc. E.A. Arsiriy[2], Ph.D. A. V. Arsiriy[3] ,

[1,2]Одеський Odessa national polytechnic university, [3]Odessa national university
Ukraine, Odessa
e.arsiriy@gmail.com

Success of real-time bidding (RTB) and advertising campaigns depends on efficient and precise user identification, audience targeting and data exchange between SSP (supply side platform) and DSP (demand side platform). Analysis of activities on the DSP side has allowed to reveal at least two problems related to information uncertainty and profiles fragmentation of users.

Keywords: online advertising, real-time bidding, demand side platform, supply side platform, audience targeting.

Today's strategic vision of Advertising (Ad) business and related technologies tightly depends on real-time bidding (RTB) and programmatic Ad technology. Modern informational technology (IT) of Ad delivery in terms of RTB can be considered as a real-time auction between Demand-Side Platform (DSP) and Supply-Side Platform (SSP) [1]. Analysis of actual sources and publications on this domain [2] allowed representing IT as a technological sequence, consisting of 10 steps, which is run while an end-user's web browser is loading publisher's web page [2]. Any interactions between DSP and SSP must conform to brand new API protocol *Open RTB* [3].

During decision making on the DSP side there are several problems related to user identification and information uncertainty. Uncertainty appears due to the concept, DSP and SSP being unfamiliar to each other, and lack to identify an end-user by *ssp_cookie* on the DSP side. DSP do not know anything about ssp_cookie. Instead, each user profile is addressed by an own DSP's special unique identifier (*dsp_cookie)* and consists of associated segments (each segment specifies a particular user's interest). The process of mining these dsp_cookies and segments mostly depends on *retargeting* activity when sites-partners of DSP (for instance, retails like Amason, news like CNN etc.) allow DSP to track end-users by connecting hidden *dsp_cookie* and passing corresponding site's content category (segments).

So the problem is once DSP has received an auction invitation it does not know anything about *ssp_cookie* and underlying end-user. Efficient mapping between *ssp_cookie* and *dsp_cookie* is one of the biggest DSP challenges besides profile fragmentation.

But, one of the biggest problems, Ad players have faced with, is profile fragmentation [4]. Nowadays almost everyone has multiple Internet entry points like tablets, phones, home PC and workstations (Fig. 1).

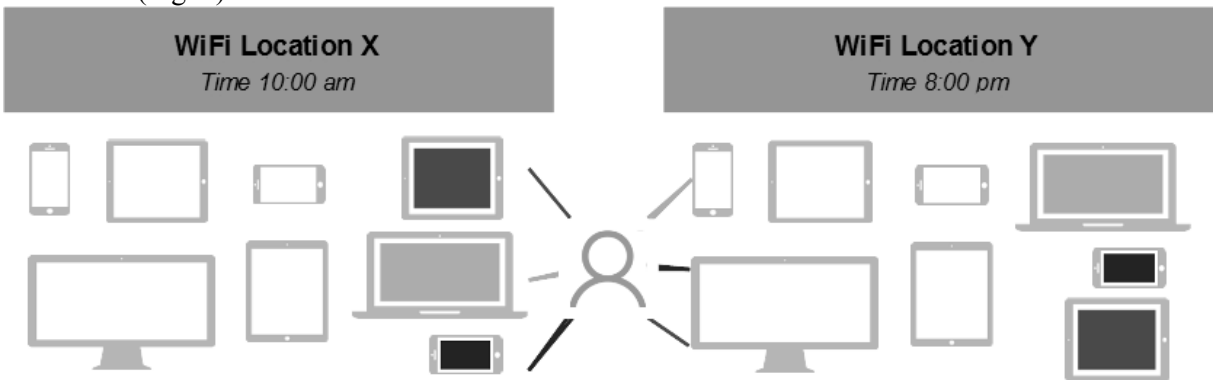

Fig. 1. Profile fragmentation problem

Every such device has own unique traceable identifier (ID). In some cases like web serving that ID might be ordinary web cookie, for mobile devices it is device ID. From Demand-Side Platform (DSP) it turns out that database has multiple profiles assigned to different IDs however connected to the same user in fact. The problem here is that it prevents from building efficient AD campaigns. For instance, two user profiles are given with own ID (assuming that the user accessed WEB via home and work PCs). The first profile provides particular user interest (segment), that he is a male. The second profile provides information that the user is a higher educational person. Splitting information about gender and education between two different profiles prevents from involving this user in complex campaigns like delivering Ad

to all males with higher education, just because from DSP perspective there profiles are two different persons. For highly-competitive Ad market that is a serious limitation.

The process of identifying profiles, connected to the same user, is called as a profile bridging. From mathematical perspective profile database is a huge graph $<V,E>$ where vertexes V are user profiles and edges E are bridging rules. Once bridging rules are well-defined then the task might be reduced easily to well-known problem of connected components identification.

The challenge is to keep profile database up to date and consistent in terms of high concurrency. Another challenge is an efficient database schema to store profiles and connections between them an a way to keep major DSP operations fast and cheap.

Graph databases: There are many specific databases to address graph problems and store data in a graph manner, like neo4j, Titan, s2graph. Just a few graph databases (e.g. Titan) are able to process graphs with billions of edges and vertexes with small response time (low latency).

For many cases Titan over HBASE is a nice choice. It is distributed graph database focused on high scalability and distributed processing. In addition, it provides modern graph API based on Blueprints interface and user-friendly query language Gremlin.

The down side of Titan is the follows:

– Titan compels own HBASE schema and the obfuscated data representation.

– It requires exclusive access to HBASE rows and columns.

– Source code of Titan provides obfuscated and complicated support of multiple HBASE versions based on shims which prevents from seamless integration with distributed computation frameworks like Scalding.

For companies that already have large HBASE profile databases with well-developed infrastructure based on Scalding/Spark integration, these issues might be critical.

Requirements and constraints: From DSP's perspective HBASE data schema and underlying data model should conform to the following requirements:

–an efficient access to user profile (corresponding segment list) by any of connected identifiers;

–an efficient check whether two profiles are connected, comparable by time to HBASE row lookup;

–an efficient retrieval of all linked profiles for a specific profile;

–simple and extensible data schema,

–Respecting super-node issue.

In addition to requirements, the schema should address the following constraints:

–minimize amount of read operations to HBASE;

–no need to implement complete operations over graph database;

–simple integration with third-party frameworks like Scalding or Spark.

Conclusions: Thus, this task of users profile fragmentation might be reduced to mathematical graphs and connected components identification algorithms. All database profiles shall be treated as vertices, while these bridging identifiers are edges. The found connected graph's components will consist of profiles linked to a single user. This gives a chance to bridge them and merge all segments into a single list. A serious question is how to build this graph and run algorithms. In case if data size is ~1B profiles, then building a graph from scratch every time might be a challenge. The corresponding graph databases like Titan [9] do not support an incremental connected component algorithm on such data scale.

SOURCES

1. Diagramming the SSP, DSP, and RTB Redirect Path.- http://www.adopsinsider.com/ad-serving/diagramming-the-ssp-dsp-and-rtb-redirect-path/

2. Trofimov B. F. User Identification Problems on DSP Side in Terms of Advertising RTB Auctions/ B. F. Trofimov, E. A. Arsiriy, A. V. Arsiriy // Information Technologies in Innovation Business conference (ITIB). – 7 – 9, October, Kharkiv, 2015. – pp. 97 – 100.

3. OpenRTB API Specification Version 2.0. - http://www.iab.net/media/file/OpenRTB_API_Specification_Version2.0_FINAL.PDF

4. B.F. Trofimov Model to represent large graphs of User profiles in hbase on DSP side / B.F. Trofimov , A.V. Arsiriy, E.A. Arsiriy// Адаптивні системи автоматичного управління. Міжвідомчий науково-технічний збірник. — Київ: Національний технічний університет України “Київський політехнічний інститут”. – 2015. – Вип. 2(27). – С. 3 – 9.