September 11, 2017

A/B Experiment Framework For Native Ads In Ad Servers

Bo Lin
Chetna McDonald
Roman Apostol
Sonny Thai
Woo Lee

See next page for additional authors

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

Recommended Citation
Lin, Bo; McDonald, Chetna; Apostol, Roman; Thai, Sonny; Lee, Woo; Shalai, Mikhail; Zhu, Zutao; and Abiri, Olawale, "A/B Experiment Framework For Native Ads In Ad Servers", Technical Disclosure Commons, (September 11, 2017)
http://www.tdcommons.org/dpubs_series/665

This work is licensed under a Creative Commons Attribution 4.0 License.
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.
Inventor(s)
Bo Lin, Chetna McDonald, Roman Apostol, Sonny Thai, Woo Lee, Mikhail Shalai, Zutao Zhu, and Olawale Abiri

This article is available at Technical Disclosure Commons: http://www.tdcommons.org/dpubs_series/665
A/B EXPERIMENT FRAMEWORK FOR NATIVE ADS IN AD SERVERS

ABSTRACT

Techniques are described for experiment frameworks around server-side rendering of native advertising content for presentation within an application environment. As described, a server receives a request, from a client device, for content that includes a content slot identifier. The server, using the content slot identifier, accesses a database to identify a presentation schema used to arrange content components and generates rendered content files according to a specific configuration for display within the environment of the resource. The server renders the content components based on the identified presentation schema and transmits the rendered content to the client device for display within the content slot of the resource. To run experiments, multiple versions of the content components and/or specific configuration parameters are used as “styles,” such that a single instance of advertising content may be presented in multiple different styles. Advertising feedback, e.g., in the form of click through statistics, conversion attributions, etc., is then used to compare the different styles and identify styles for further use.

DETAILS

Data and communication networks, particularly the Internet, and more particularly the use of these networks for distribution of media content, have great potential for improving the delivery of content to receptive audiences. Content may include text, images, audio, video, multi-media, interactive media, games, applications, and any other content that can be delivered via a network. Content may include advertisements (“ads”), which may be any form of communication in which one or more products, services, ideas, messages, people, organizations or other items are identified and promoted (or otherwise communicated). Ads are not limited to
commercial promotions or other communications. An ad may be a public service announcement or any other type of notice, such as a public notice published in electronic press or a broadcast. An ad may be referred to as, or include, sponsored content. For example, some websites provide information search functionality that is based on keywords in a query entered by a user seeking information. This query can be an indicator of the type of information of interest to the user. By comparing the query keywords to a list of keywords specified by an advertiser, it is possible to provide ads relevant to the user. Another form of online advertising is ad syndication, which allows advertisers to extend their marketing reach by distributing ads to additional partners. For example, third party online publishers can place an advertiser’s text or image ads on web properties with desirable content to drive online customers to the advertiser’s website. Some ads are classified as Native Advertising.

Advertising content that is integrated into its presentation context in a cohesive manner, e.g., assimilating application environment design elements, such that the overall presentation experience is consistent and generally uniform can be referred to as “Native Advertising.” Native content can include, for example, content having a similar form and function as primary content of a presentation context into which the native content is to be inserted. The presentation context may be presented by an application (such as a web browser, a media streaming application, a game, a mobile “app,” etc.) executing on a client device (such as a desktop computer, a portable laptop or notebook computer, a tablet, a mobile “smart” phone, etc.), and the native content may appear integrated into the presentation of the application. Native advertising content is different from other forms of advertising in that it generally appears to blend into or “fit” in the environment or context in which is presented. This natural presentation can be more pleasing and comfortable for the audience without being deceptive. Indeed, the
Interactive Advertising Bureau ("IAB") recommends that, for paid native advertising content, there should be a clear and prominent disclosure that conveys that the placement of the native advertising content has been purchased. (See, e.g., https://www.iab.com/guidelines/native-advertising/)

Native content can be rendered by the client device, and presented by the application within an environment of the application. For example, a raw content asset including one or more content components may be transmitted from a server to the client device, where the application at the client device uses specialized rendering tools, scripts, or instructions to assemble the content components of the raw content asset according to predetermined parameters of the application. Once the application renders the native content using the content components, the application presents the rendered native content for display within the environment of the application such that the rendered content appears to have the same look and feel as publisher-generated content.

Native content can be rendered by a content server, and delivered in a rendered form to the client device for presentation by the application within an environment of the application. An example method for server-side rendering of native content for presentation within an environment of an application includes receiving, by a data processing system, from a client device, a request for content including a content slot identifier identifying a particular content slot provided by a resource. The content slot identifier is then used by the server to determine a presentation schema according to which to assemble, arrange, style, and/or present a content item within the content slot identified by the content slot identifier. Characteristics of the content item may be made to substantially match characteristics of other content included in the application.
(e.g., font, text size, background color or pattern, etc.). An example data processing system for delivering rendered native content is shown in the following illustration:

![Data Processing System Diagram]

The data processing system shown above might be used, for example, as a content placement system. It may be implemented on a single server or implemented across multiple servers, e.g., distributed across devices in a data center or server farm. The illustrated data processing system includes a database, which may store web pages, portions of webpages, third-party content items such as advertising content, scripts such as a prefetching script, and so forth. The illustrated data processing system includes three modules: a content request module, a content rendering module, and a rendered content delivery module. The content rendering module may be further divided into a conversion module, a content selection module, a presentation lookup module, a presentation filter, a ranking module, and so forth. One example of a content rendering module is illustrated below.

The content request module, the content rendering module, and the rendered content delivery module can include software components, hardware components, and/or combinations of software and hardware. For example, the modules may include one or more processors configured to execute one or more scripts to obtain requests to initiate a content item placement, identify information resources of a content publisher computing device, provide a rendering of the information resource at the content provider computing device, identify content item slots of
the information resource, provide content items for display in the rendering of the information resource, or receive authorization to provide content items for display with the information resource by an application at a client device.

The content request module can receive a request for content from the client device. The request for content may include a request for an information resource, a request for one or more third-party content items, or a combination thereof. In some instances, the request for content may include a request for third-party content, which may include an address or identifier of an information resource on which the third-party content is to be displayed (e.g., a web page presented in a browser, a page of an electronic document, an interface presented within a mobile device “app,” etc.). The request for third-party content can also include or identify one or more parameters that can be used by the data processing system to determine the content to provide in response to the request for content. For example, the parameters can identify a size of a content slot within which to insert the requested content. The parameters can identify a type of content associated with the information resource, a type of third-party content requested (e.g., text, image, video, etc.), client device information, size information for requested third-party content item, a network type associated with the client device (e.g., cellular network, WiFi, etc.), an application ID associated with an application executed by the client device, attributes associated with the application ID, a current or past location of the client device, or a combination thereof. The request may include an address or identifier of an information resource. For example, the request for the information resource can include a Uniform Resource Locator (“URL”) of a specific resource such as a webpage (e.g., “http://www.example.com/”). The request for information resource can also include client device information (such as a device type, device identifier, or a combination thereof).
The content rendering module selects a content item for rendering and delivery to the client device. The content rendering module can access data stored in one or more databases, e.g., a content asset database, a presentation schema database (storing presentation schemas that each dictate how content items should be rendered), a presentation modification or experimentation “style” database (storing parameters for variations in presentation, e.g., for experiments), a scripts database, etc. The various databases may be separate distinct databases, or one or more may be portions of a same database. For example, the content rendering module may access a database for selection of the content item.

In some instances, the content rendering module accesses a presentation schema lookup database for selecting a presentation schema that dictates how the selected content item should be rendered. The content rendering module renders the selected content item based on the selected presentation schema. The rendered content item may include a native content item to be displayed within a content slot of an information resource such that the rendered content item substantially matches visual characteristics of an environment of the information resource. In some instances, an experiment framework includes multiple “styles” for rendering the native content, such that the content rendering module may render the same native content in different ways for presentation in the same or similar contexts. A content item may be provided to a first client device (“A”) for presentation in a particular context and provided to a second client device (“B”) for presentation in the same particular context, but formatted in a different style (e.g., presentation at device “A” may be in a different format or layout than presentation at device “B.”)

In some versions, the content rendering module includes a content manager that identifies a context presentation schema by selecting a presentation schema compatible with the request.
For example, the request may include a content slot identifier and size information, and the content manager selects a scheme compatible with the identified content slot and size information. The content manager may select a list of content assets based on the content slot identifier, select a list of presentation schemas assigned to, or otherwise associated with, the content slot, and filter out incompatible content assets, such as those that do not include corresponding presentation schemas matching the size information.

In some versions, the content rendering module includes a content selection module that accesses a content asset database to select a list of native content assets responsive to the content request. For example, a content asset provider may assign or link a particular content asset to one or more content asset templates, and the content asset template can include a specification according to which content components of the content asset can be arranged. The content components can include, but are not limited to, a headline, an image, a body, a call to action, a logo, an attribution, a click through URL, or the like. The content selection module may use a query to identify content assets from the content asset database, e.g., a query including the content slot identifier and the size information from the content request. The content asset database can match, map, or link content assets assigned to a particular content asset template compatible with a presentation schema assigned to a content slot. The content asset database may also maintain sizing information of content assets and map each of the stored content assets with the sizing information. As such, the content selection module can perform a lookup in the content asset database based on the requested content slot identifier and the size information, and retrieve corresponding content assets.

The rendered content delivery module forwards the rendered content from the content rendering module to the client device for delivery. For example, the rendered content item may
include an identifier that allows the rendered content delivery module to determine a particular
destination for transmitting the rendered content item. The identifier can be received by the
content request module along with a request for content from the particular client device.

The rendered content delivered by the rendered content delivery module can be
configured with an interaction script. In some instances, the interaction script includes
computer-executable instructions that cause the client device to transmit an indication of an
interaction with the delivered content. For example, the interaction can include a click, swipe,
tap, hover-over with a cursor, or any other interaction with a content item. The computer-
executable instructions, (such as HyperText Markup Language (HTML), Extensible HyperText
Markup Language (XHTML), Extensible Markup Language (XML), Cascading Style Sheets
(CSS), JavaScript, etc.) can be executed within an application or web browser at the client
device, e.g., the application or web browser that generated the content request received by the
content request module. The indications reported by the interaction script may be used as
feedback to assess value of or interest in the content. In some instances, content value may be
determined using outside means, such as later conversion reporting or measured increase in
related activity.

To test variations in content, the content rendering module may be configured to deliver
different renderings of the content to different client devices. The assessed value of or interest in
the content may then be compared based on indicators from client devices receiving the different
renderings. This is described in more detail below, in reference to the accompanying flowcharts.

The following flowchart illustrates a method of providing native content to client devices
rendered according to experiment styles:
In the flowchart shown above, a data processing system (e.g., for a content placement system) receives (from a client device) a request for content. The system, responsive to the content request, identifies a context presentation schema, identifies an experiment presentation style (e.g., when there is an experiment active), and identifies content assets. The system renders the content assets in accordance with the identified schema and experiment presentation style, and provides the resulting rendered content to the client device responsive to the content request.

As introduced above, the requested content may be constructed from multiple content assets, including native advertising content. The request may indicate, or otherwise be associated with, a particular presentation context. In some versions, the content rendering module includes or accesses a database of context presentation schemas, and selects an schema based on the received request for content. In some versions, the schema is a cascading style sheet (“CSS”). In some versions, the schema is a web page template. The schema may indicate presentation qualities such as a color palette, a layout, fonts and font sizes, and so forth. The content rendering module renders the identified content assets in accordance with the selected schema.

A content provider may configure an experiment, wherein certain content assets (e.g., native advertising content assets) may be presented in various styles. The content provider may then compare reception statistics associated with the various style presentations. In some versions, the content rendering module includes an experiment control module that regulates
presentation of the various style options. The experiment framework collects audience feedback, e.g., in the form of click through statistics, conversion attributions, etc., which may then be used to make decisions about future utility of the content presentation styles.

After presenting the rendered content to the client device, the data processing system may obtain feedback from the content recipient. This audience feedback may be received in the form of client-side interaction (e.g., mouse hover, click, interactive selections, media play completion, accessing a promoted webpage or network resource, and so on). In some versions, audience feedback is measured based on resulting conversion events, such as purchases of an advertised product or service, for example.

![Flowchart](image)

As shown in the above flowchart, the data processing system receives audience feedback and correlates the feedback to the experiment presentation style. The data processing system then updates data associated with the presentation experiment based on the correlated feedback. The data processing system may update a database with feedback data (e.g., clicks and/or conversions) based on the style of the associated content delivered. The correlation may be premised, for example, on a consistent identifier, e.g., an identifier provided to the client device with the rendered content and returned by the client device in the feedback. In some instances, the consistent identifier is provided by the client device in the content request and in the later feedback. In some versions, the correlation is premised on client attributes, such as device attributes, geolocation data, time data, and so forth. This framework can then be used for side-by-side comparisons of different presentation styles for the content. In some versions, the
framework provides performance data as feedback or as actionable insights to external users. These insights can be for understanding native style performance.