An Open Source Advertisement System

Pushkar Umaranikar, Chris Pollett

Abstract—An online advertisement system and its implementation for the Yioop open source search engine are presented. This system supports both selling advertisements and displaying them within search results. The selling of advertisements is done using a system to auction off daily impressions for keyword searches. This is an open, ascending price auction system in which all accepted bids will receive a fraction of the auctioned day's impressions. New bids in our system are required to be at least one half of the sum of all previous bids ensuring the number of accepted bids is logarithmic in the total ad spend on a keyword for a day. The mechanics of creating an advertisement, attaching keywords to it, and adding it to an advertisement inventory are described. The algorithm used to go from accepted bids for a keyword to which ads are displayed at search time is also presented. We discuss properties of our system and compare it to existing auction systems and systems for selling online advertisements.

Keywords—Online markets, online ad system, online auctions, search engines.

I. Introduction

ONLINE information today is largely "free" because of advertisements shown on web pages. This is possible because more money is spent on online advertising than any other source of advertising except television [17] [6]. To provide these ads, an online advertisement system is often used. An online advertisement system is a web application that provides a marketplace and a distribution platform for advertisers to purchase and display advertisements on a collection of web pages either of a single website or a collection of websites belonging to a network. Despite enabling free access to information, these advertising systems themselves tend to be proprietary in that their implementation details cannot be freely examined by the advertisers. In this paper, we describe a new, open source, online advertisement system built into the Yioop Search Engine project that attempts to address this as well as provide novel features not currently provided by existing systems.

An online advertising system needs to provide a mechanism for pricing advertisements and a basic unit of pricing. One early pricing unit used by online advertisement systems was to sell ad space on a cost-per-thousand-impressions. As advertisers sought better signals of buyer interest, and hence, likelihood to buy, newer systems began using auctions based on pay-per-click or pay-per-engagement. Traditional print or television media, in contrast, tend to sell ads based on units of time, for example, thirty second advertisements shown for some number of days or a print ad shown for some number of days in a classifieds section. Sites like `Craigslist.org` illustrate this classified model to some degree in the online setting.

We call a system **pay-per-slot** if the advertiser pays for their advertisement to appear during a certain time slot without a guarantee that the ad is seen or clicked on in that time slot. Good analytics allow one to estimate how likely a click, an impression, or time slot is going to be converted to a sale. For example, knowing the average number of times a query is done per time slot allows one to convert cost-per-slot to cost-per-impression; knowing the average number of clicks-per-impression allows one to convert the latter to cost-per-click. Finally, one can try to estimate either from impressions or clicks a conversion rate. Although the error in these estimates might be lower for whether a click converts to a sale, pay-per-slot does offer benefits in certain situations. To see this, consider the release of a new movie where marketing to distributors and the opening weekend box office largely determines the overall gross of the movie [10]. In such a situation, one might be willing to sacrifice number of conversions accuracy for the ability to precisely control when and where the advertisement appears. Pay-per-click systems often choose ads to display according to remaining budgets of advertisers for a keyword according to an online algorithm such as the BALANCE algorithm [5]. As the search engine generates revenue only if an ad is clicked, this means an ad unlikely to be clicked will tend to be passed over by an online advertisement system for display in favor of an ad that is. This in itself creates uncertainty in the reach of ones ad at the time one wants. It also creates an incentive for "click bait" ads over awareness building ads which can lead to user dissatisfaction with both the search engine and the advertiser.

In the Yioop advertisement system described in this paper, a pay-per-slot system is used to sell advertisement space. Users have the option of specifying a duration for their ad campaign. Ads targeting a keyword are displayed based on the fraction of the total expenditure for that term and time slot. Pricing of slots is not via a traditional auction. Instead pricing starts at 1 ad credit for the first bid on a given time slot. Each additional bid on the time slot must be at least the maximum of 1 ad credit or one half the total of the bids so far on that slot. The latter condition ensures that there are at any time at most logarithmically in the total ad spend many ads that might be displayed at query time for a time slot. The value of an ad credit in a usual currency such as dollars can be determined by a Yioop site owner to reflect the total traffic that their website is likely to generate.

In the rest of this paper, we develop how the above system is implemented in Yioop and how it compares to existing pay-per-click systems. Section II describes the Yioop search engine that our ad platform was integrated with. We then in Section III describe the Bing and Google Ad platforms. We

Pushkar Umaranikar is with Rackspace, 1 Fanatical Pl, San Antonio, TX 78218 (e-mail: pushkarumaranikar1@gmail.com).

Chris Pollett is professor in the Department of Computer Science, San Jose State University, 1 Washington Square, San Jose CA 95192 (e-mail: chris@pollett.org).

then discuss how ad selling auctions work. Section V describes the implementation of our online advertising system. Section VI compares our system to the systems described earlier and gives some experimental results. Finally, the last section gives some closing observations on our system.

II. Overview of Yioop

The Yioop Open Source Search Engine project was started by the second author, Chris Pollett, in 2009 with its first public release in 2010 [19]. Several features make it attractive for experimenting with new search related technologies: It comes out-of-the-box with a search engine, social media and wiki platform. It is written in a scripting language PHP which makes suitable for rapid experimentation as compared with many compiled languages like Java or C++. It does not rely on third party libraries which can make such search and ad systems hard to deploy. It has been used in web-scale crawls of up to a billion pages and has been used in several master's student project in computer science at San Jose State University. Since our online advertising system displays keywords advertisements within search results, Yioop software serves as a convenient starting point.

III. Online Advertising Systems

Before we discuss the details of our new online advertising system we would like to review the features of two popular systems, Bing Ads and Google AdWords. We would also like to briefly discuss ad networks, so that we may compare them with ours later in the paper.

A. Bing Ads

Bing Ads is an advertising service used by both the Bing, Yahoo, and AOL. Originally, Overture and then Yahoo provided ads for Bing's precursor, MSN Search. This Overture connection pre-dates Google AdWords which we describe later. Starting in 2006, Microsoft developed its own advertisement platform, which has been improved internally and via acquisitions since then. In 2010, Yahoo and Microsoft reached an agreement that Bing would provide both search and search ad results for Yahoo. This was followed in 2015 by an agreement between AOL and Microsoft that resulted in AOL taking over Microsoft's video, mobile, and display ad business, and in Bing and Bing Ads being used in AOL's search services [3]. As described in its documentation [15], two factors factors are involved in determining if an ad gets displayed by Bing. The first factor is a function of the amount an advertiser is willing to pay for a click combined with the estimated click-through-rate of the ad. The second factor is what words and matching criteria the user is targeting. Bing Ads supports broad, phrase, exact, and content matching criteria.

Selecting broad match allows an advertisement to be displayed if any of the ad keywords appear in a search query, in any order. Using phrase match allows an ad to be shown if all of the words in a key phrase match the words in a search query, in exactly the same order. There can be other words in the search term. Exact match permits an ad to be shown when the exact words in your key phrase appear in a search query, in exactly the same order. Content match ads are ads that are placed in a websites external to Bing, but are part of Bing's ad network.

B. Google AdWords

Google started their online advertisement platform in 2000, known as Google AdWords [7]. This was then overhauled greatly in 2002. One of the main services of AdWords is to place advertisement copy in the space available on Google's search results page. AdWords offers cost-per-click (CPC) and variants of cost-per-click advertising. On non-search result pages, they also offer cost-per-thousand-impressions, also known as cost per mille (CPM), advertising. AdWords also provides site-targeted advertising for text, banner, and rich-media ads as well as re-marketing.

The user level details of AdWords are similar to Bing Ads. AdWords allows users to create advertisements, associate these ads with a given set of keywords, choose a maximum cost-per-click, and a daily budget. [8]. On a given Google query, AdWords shows chooses an advertisement to display from amongst those involving the query keywords, based on a function of its likely click-through rate, its remaining ad budget for the day, and its max per-click budget [5].

C. Ad Networks

As we mentioned above, Bing Ads supports content-match ads which are displayed on its external ad network. Similarly, Google AdSense allows website owners to earn revenue from their online content. We spend a moment to describe this and ad networks in general since as part of the advertisement enhancements we made to Yioop, we added the ability for Yioop site owners to make use of an ad network.

AdSense works by matching images and text to the owner's site based on content and visitors [16]. To display advertisements, website owners place a Javascript code provided by AdSense on web pages.

Advertisements displayed by Google AdSense as well as by other ad networks can contain an advertising icon in the corner of the advertisement known as AdChoices. This icon's purpose is to inform users that information about user's interest is being gathered to improve advertisements displayed for that user [14].

IV. Using Auctions to Sell Advertisement Spaces

Continuing our discussion of existing ad systems before describing the system implemented in Yioop, we now consider the most common ways to go from a list of ads relevant to a query, each with its own likely click-through rate and maximum allowed cost-per-click, to a selection of ads to display. In systems such as AdWords and BingAds this is done through an auction-like mechanism. Following Krishna [9], we define an **auction** to be any method of selling goods or services where bids are elicited from the buyers or sellers and the goods and services sold based on the received bids. We require that the mechanism for choosing the winning bidder or bidders be

universal in that it does not depend on the particular goods or services being sold and that the mechanism be **anonymous** in that it does not depend on the properties of the bidders beyond the bids. In our context, both the click-through rate and the maximum allowed cost-per-click are viewed as part of the bid. If one does not view the click-through rate as part of the bid these ad system would probably not meet the definition above. Common metrics used to judge auction systems [2] are:

1) **Truthfulness in Bidding** – that is, bidders bid their true value for the item, not some strategic value.
2) **Seller's Revenue** – the amount of revenue the seller receives for the sale of the items.
3) **Bidders' Payoffs** – the sum of the payoffs received by all bidders for the items won.

Another important criteria not listed above, but which factors in the decisions of an advertiser to use a platform or not is the **uncertainty in the bidder payoff** – an advertiser might choose a system with a lower payoff if the uncertainty in obtaining that payoff is lower.

The two most common types of auction are open and sealed envelope auctions.

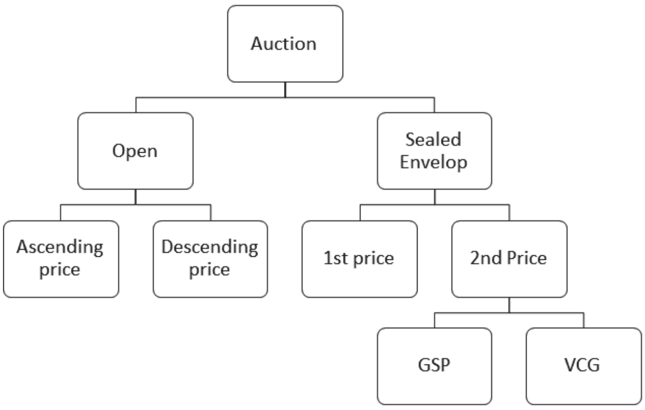


Fig. 1 Auction Methods

Open auctions are perhaps the most familiar type of auction to the average person. They might be used in the sale of art or collectibles in an auction house like Sotheby's or in the sale of items on eBay. Open auctions are conducted at a public venue where advertisers announce their bids. There are two types of open auction: Ascending price, and descending price. In an ascending price auction, the auctioneer may or may not start at some reserve price and asks for bids. A bidder can bid a value greater than the maximum bid so far. The auction continues to receive bids until one bidder submits a bid that no other bidder is willing to exceed. The last bidder wins and pays the last announced bid price. In a descending price auction, an auctioneer starts with a high price. The price is meant to be so high that no bidder would be willing to accept it. The auctioneer then lowers the bid amount until one of bidders agrees to pay the announced price. This bidder wins and pays the last price announced by the auctioneer [2].

An alternative kind of auction to an open auction is a sealed envelope auction. In this kind of auction each bidder submits bid privately. All bids are revealed to the auctioneer who then decides the allocation and charges for each advertisement space. The highest bidder gets the advertisement space. The amount charged depends on which auction method is used, usually highest or second price bid.

In a first price auction, the winner pays the highest bid. In a second price auction, the winner pays the second highest bid. Naively, one could imagine that if a bidder knows that he or she is paying the second highest bid, the bidder would bid an extremely high price. Thus, if the bidder wins, the bidder pays this much lower price. However, a bidder can't be sure that other bidders will not try to follow the same strategy, so it is safer, all things being equal, for a bidder to follow a truthful strategy.

In the case where one item is being sold, it is known that descending price open auctions are equivalent to sealed first price auctions, and that ascending price, open auctions are equivalent to second price, sealed auctions. It is also known that the open and sealed auctions described above encourage truthful strategies in bidding. From a practical standpoint when selling ads, the other advertisers only have limited knowledge of who will bid on a given keyword at a given time, and so in designing an auction system for ads, it makes sense to work in the sealed bid setting.

Often when selling advertisement space, there are several ad slots available on a search result page. To handle this, a generalized second price (GSP) auction is used. Here the top slot goes to the highest bidder at the second higher bid price, the second slot goes to the second highest bidder at the third highest price, and so on. Although easy to describe, there are known cases where GSP leads to untruthful bidding. A Vickrey-Clarke-Groves (VCG) auction is a modified form of generalized second price auction where we try to get a maximum valuation for the matching of sellers to bidders which is known to encourage truthfulness. However, as it is more complicated to describe, it is not typically used by ad networks.

One quirk of online ad auctions is that often several advertisers for a query may have the same, or nearly the same, expected cost-per-click. Here the expected cost-per-click is the product of the advertiser's bid and the estimated click-through rate based on prior ad shows. To maximize revenue, an ad system wants to maximize the cost-per-click across all queries. This usually involves using the remaining daily budget of an advertiser to determine if they get to participate in a given ad auction. Variations in the function of bid, click-through-rate, and remaining budget used to figure out who participates in an ad auction can be used to optimize the auction for ad-quality, clicks, buyer profit, conversions, or revenue [4].

V. The Ad System in Yioop

We are now in a position to describe the ad system we developed for the Yioop search engine. We split our description into steps. First, we explain how it works operationally for the end-user, then how it was implemented, and finally how the auction system works and we compare it to existing ad auctions.

A. The Business User Role

The Yioop search engine uses role-based access control to determine which activities a given signed-in user has access to. The first extension made to Yioop for the ad system was to create two new activities Manage Credits and Manage Advertisements as well as a new role “Business User” that gives a user access to these activities. Not all users of Yioop will want to buy advertisement, so to reduce clutter on an average user’s account pages a standard user is not given the Business User role. On the other hand, we do want any user to have the ability to become a business user, so we added a toggle under Manage Accounts to enable this role. Roles are administrator configurable in Yioop, so it is also possible through the GUI interface to configure the default User role so that it allows the Manage Credits and Manage Advertisements activities without the need of the Business User role.

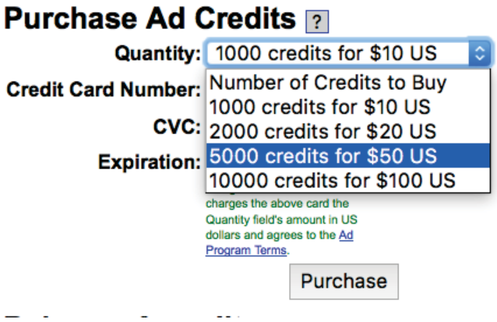
Fig. 2 Advertisement Component

B. Managing Credits

In a relatively light internet traffic setting, unlike the setting of a major search engine, there may be only a few thousand or hundred of thousands of unique visitors a day. As keyword advertising in this setting has less reach, the value of an advertisement will tend to be lower than in the large scale search setting. Our ad system uses Ad credits to allow users to purchase ads for effective rates that might be as low as a penny per day of advertisement, but also minimizes the fees that the site owner of a Yioop instance has to pay to a credit card company for a given transaction. As can be seen in the figure showing the Purchase Ad Credits form, advertisers can purchase between 1000 and 10000 ad credit in bulk at a cost of between $10 and $100. Once the advertiser has purchased credits, these can then be used to purchase keyword advertisements in our as ad auction system. Purchases are done through credit card using `stripe.com` for processing. Since the minimum credit card transaction is for $10, default minimum transaction fees that this processor has are avoided. The precise rate of credits and units purchased can be adjusted to reflect what advertisers are willing to pay for keyword ads in the system and minimum fees that a given payment processor might have. Beneath the Ad Credit form in the Manage Credit activity is a searchable and pageable ledger maintaining a list of credit transactions as well as purchases of ads using credits.

C. Managing Advertisements

To create an advertisement in our ad system, one uses the Purchase Ad form shown in Fig. 4. The fields in this form include Ad Title, Ad Body, Destination URL, Campaign Duration, Keywords and Budget. In the Budget field, the user has to enter an amount that is equal to or greater than the minimum required budget. This minimum required budget value is calculated by the auction system depending upon the number of keywords and their popularity – we will describe the exact mechanism in a later subsection.


Purchase Ad Credits ?

Quantity: 1000 credits for $10 US

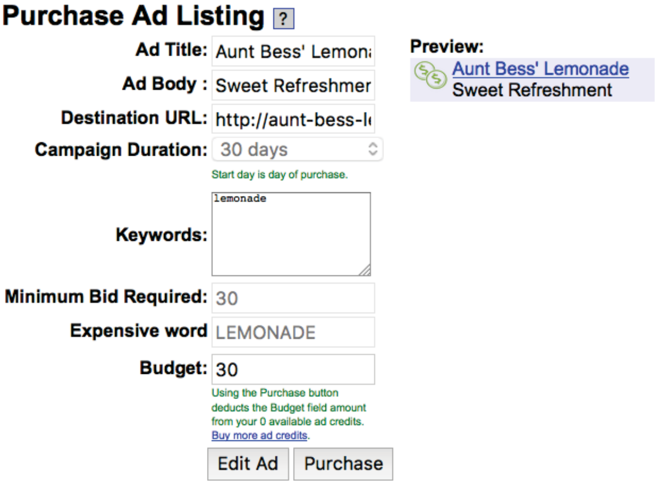
Credit Card Number:

CVC:

Expiration:

charges the above card the Quantity field's amount in US dollars and agrees to the Ad Program Terms.

Fig. 3 Purchasing Ad Credits


Purchase Ad Listing ?

Ad Title: Aunt Bess' Lemona

Ad Body : Sweet Refreshmer

Destination URL: http://aunt-bess-l

Campaign Duration: 30 days

Start day is day of purchase.

Keywords: lemonade

Minimum Bid Required: 30

Expensive word LEMONADE

Budget: 30

Using the Purchase button deducts the Budget field amount from your 0 available ad credits. Buy more ad credits.

Preview:

Aunt Bess' Lemonade
Sweet Refreshment

Fig. 4 Creating an Advertisement

Along with the minimum budget required, the auction system displays expensive words among the keywords associated with an advertisement. It helps users to manage their budget. User can try minimizing their budget by removing expensive words from the keywords list.

Beneath the Purchase Ad form is a searchable table display current and previous user ad campaigns. This table provides useful analytics on the number of impressions and clicks a given ad has received. User can edit, deactivate and activate already created campaigns. Advertisements are automatically deactivated on the expiry date. Once an advertisement is deactivated, a user can no longer edit it. When editing an existing ad, only the Ad Title, Ad Body, Destination URL are editable.

D. Storing Credit and Advertisement Information

The Yioop search engine uses its own custom format to maintain web indexes, but keeps all of its user, group, wiki, dynamic localizations, authentication, and access control information in a traditional database management system (DBMS). Yioop has been tested against three kinds of DBMS’s: Sqlite, Mysql, and Postgresql, although it should work with other systems with relatively minimal changes. The default DBMS is Sqlite which stores whole databases in single binary files. Given how locking works in Sqlite, it is better suited for single user settings or when the DBMS is intended to be mainly read. Keeping track of user credits and ad campaigns involves many transactional queries, so in a production environment it would be better to use a database that supports transactions such as MySQl or PostgreSQL. As we mentioned above, ad credit purchases are done using `stripe.com` to handle the credit card transactions. This avoids ever having to store credit card information in the Yioop system or on the Yioop servers – the browser of an advertiser making a credit purchase makes a secure Ajax request to `stripe.com` with the Yioop instance's Stripe public key to get a security token, this token and the ad credit purchase details are then sent to the Yioop server, and finally, the token and the charge amount are sent from the Yioop server to `stripe.com` to actually carry out the charge. Although credit card information is not stored on the Yioop installation, it is still desirable that all the ad credit forms and ad listing forms be served using https. The portion of the Yioop database schema used to manage ad credits and ad listings is relatively simple consisting of three tables as seen in Fig. 5. We briefly describe the intended purposes of these tables below:

1) ADVERTISEMENT
   Contains information about advertisements including metadata such as clicks and impressions.
2) ACCEPTED_AD_BIDS
   Contains keywords and associated bid amount with each keyword.
3) CREDIT_LEDGER
   Maintains information about user credits.

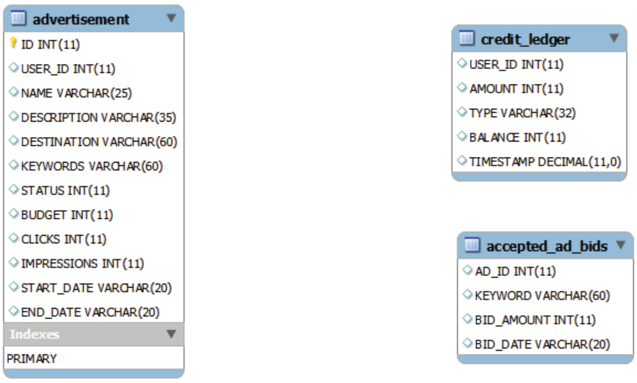


Fig. 5 The Database Schema for the Yioop Ad System

E. Calculating the Minimum Bid Amount

Our database scheme described above could support several different bidding mechanisms. In this section, we describe how our auction system calculates the minimum bid required to create a new campaign based on the popularity of keywords. As discussed in the introduction, we will describe in the next section how our auction system is compared to existing systems. Auctions in our system are for the opportunity to have ones ad displayed associated to a keyword for a whole day. When a user purchases an ad associated to multiple key phrases for a duration of more than one day, each day in the duration and each key phrase on that day is treated as a separate auction. The sum of the costs of all these auctions is used to determine the bid. As an example, suppose a user is creating a new campaign entitled "Test Advertisement". Suppose further that the keywords associated with this campaign are "test" and "advert" and that the duration of the campaign is one week. Consider Tables I and II describing the existing total bid amounts associated with each keyword.

TABLE I
Bid Amounts Associated with the Keyword "Test" for Each Day

Bid Day	Total Bid Amounts
1	10
2	8
3	10
4	12
5	10
6	8
7	0

TABLE II
Bid Amounts Associated with the Keyword "Advert" for Each Day

Bid Day	Total Bid Amounts
1	8
2	12
3	10
4	12
5	8
6	4
7	12

Tables II and IV show the current total bid amounts for the keywords "test" and "advert" on each day of the campaign. Using these bid amounts, the auction system calculates the minimum bid required. As briefly described in the introduction, to calculate the minimum bid, we take the maximum of half the existing bid total for each day and one for each keyword. We chose a half as it is fraction that can easily calculated as a shift on a computer. The usefulness of choosing a constant fraction of the total bid amount will be shown in a later subsection. So, the bid amount for the keyword "test" on Day 1 is 10. Half this bid amount is 5 which is greater than 1. So the minimum bid for this portion of the campaign is 5. On Day 7, the keyword "test" has not been bid on yet. Hence, the bid amount for "test" on that day is 1.

The minimum bid for the word "test" is thus:

$$\begin{aligned} &= (10/2) + (8/2) + (10/2) + (12/2) + (10/2) + \\ &\quad (8/2) + 1 \\ &= 5 + 4 + 5 + 6 + 5 + 4 + 1 \\ &= 30. \end{aligned}$$

TABLE III
NEW BID AMOUNTS ASSOCIATED WITH THE KEYWORD "TEST" FOR EACH DAY

Bid Day	Total Bid Amounts
1	(old total) + (new bid) = 10+10=20
2	8+8=16
3	10+10=20
4	12+12=24
5	10+10=20
6	8+8=16
7	0 + 2 = 2

TABLE IV
NEW BID AMOUNTS ASSOCIATED WITH THE KEYWORD "ADVERT" FOR EACH DAY

Bid Day	Total Bid Amounts
1	(old total) + (new bid) = 8+8=16
2	12+12=24
3	10+10=20
4	12+12=24
5	8+8=16
6	4+4=8
7	12+12=24

Similarly, the minimum bid required for word the "advert" is:

$$\begin{aligned} &= 4+6+5+6+4+2+6 \\ &= 33. \end{aligned}$$

Hence, the minimum bid required to create the campaign = 30 + 33 = 63. The expensive word in this case is, "advert".

F. Bidding over the Minimum Bid

In the above, when the user enters a budget greater than 63, the auction system creates a campaign. Suppose a user enters a larger value, for example, say a budget of 126. We next describe how the additional money above 63 is split among the seven days and two keywords.

Our campaign duration is seven days and we have two keywords associated with the campaign. Let α be the ratio:

$$\frac{\text{Min Keyword Bid For Given Day}}{\text{Min Bid Required For Whole Campaign}}$$

The following equation says how much will be bid for a given keyword for a given day.

$$\text{New Keyword Bid For Day} = \alpha \cdot \text{Budget}$$

As an example, for the keywords "test" on Day 1 the new bid is:

$$\begin{aligned} &= ((10/2)/63) \cdot 126 \\ &= 10 \end{aligned}$$

Using the above formula, we can fill in the new bid amounts and total bids for "test" asin Table III. Similarly, we can fill in the new bid amounts and total bids for "advert" as in Table IV.

When calculating the product $\alpha \cdot$ Budget we might end up with fractional values. To solve this problem, as we proceed through the days of the campaign for a given keyword, we round down the fractional values, but also keeps track of the sum of the left over fractions so far, if this sum ever exceeds one ad credit, then one ad credit is added to that day's bid, and the left over is deducted by one.

G. Relations between Bids and the Total Bid

One of the features of our bidding system is that the number of bids for a given keyword will tend be small. This will help ensure that the database queries to serve ads will be fast. To see this more formally, let $B(n)$ denote the nth bid for a keyword for a given day, let $T(n)$ denote the total bid amount after the nth bid.

Theorem 1: For $n \geq 1$, we have

$$3 \cdot B(n) \geq T(n) \geq (3/2)^{n-1}.$$

Proof: The minimum bid requires $B(n) \geq (1/2) \cdot T(n-1)$, so $2 \cdot B(n) \geq T(n-1)$. By definition $T(n) = B(n) + T(n-1)$. The first inequality follows from these two statements. We prove the second inequality by induction. When $n = 1$, $T(1) = B(1) \geq 1 = (3/2)^0$, so the inequality holds. As our induction hypothesis, assume the inequality holds up to $n - 1 > 0$, that is, $T(n-1) \geq (3/2)^{n-2}$. Then $T(n) = B(n) + T(n-1)$, and as $B(n) \geq (1/2) \cdot T(n-1)$,

$$\begin{aligned} T(n) &\geq (1/2) \cdot T(n-1) + T(n-1) \\ &= (3/2) \cdot T(n-1) \geq (3/2)(3/2)^{n-2} = (3/2)^{n-1}. \end{aligned}$$

Thus, the induction holds, completing the proof. ■

From the second inequality we immediately have the following corollary:

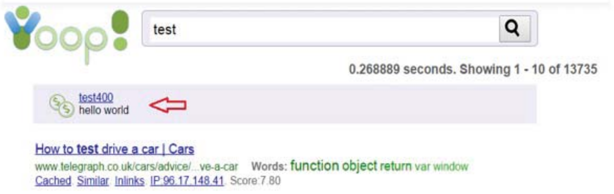
Corollary 1: If T is the total bid amount on a keyword for a given day, then the number of bids on that keyword for that day is at most $\log_{3/2} T + 1$.

H. Displaying Advertisements on the Search Results Page

We next describe how to go from a search for a given keyword or phrase and a sequence of bids that have been made on that keyword for the current day to an actual choice of the advertisement to display. Suppose the user enters the search query "computer". In addition, to finding search results for this query, a Yioop instance with our ad system enabled would look up using the ACCEPTED_AD_BIDS and ADVERTISEMENT tables all of the ads for this keyword for this day. As the KEYWORD and BID_DATE columns of ACCEPTED_AD_BIDS have an index on them, and in view of Corollary 1, this will be a relatively fast operation. As an example, suppose there are five advertisements matching the search query "computer" with budgets given by Table V. Notice the ads are sorted from least to maximum bid. The total of all the bids is 10 + 20 + 30 + 40 + 50 = 150. Using a pseudo-random generator a number x between 1 and 150 is calculated. In a loop starting with a sum at 0, we add the bid values from smallest to largest. The first ad that causes the sum-so-far to equal or exceed x is the chosen ad for display. So if x was the 25, then we would add $0 + 10 + 20 = 30$, a value larger than 25, and hence, Ad 2 would be displayed. If x were 150, the last ad would be chosen. An example of a displayed advertisement is shown in Fig. 6.

TABLE V
RELEVANT ADVERTISEMENTS WITH THEIR BUDGET VALUES

Relevant advertisement	Bid Amount
Ad 1	10
Ad 2	20
Ad 3	30
Ad 4	40
Ad 5	50



test

0.268889 seconds. Showing 1 - 10 of 13735

test400
hello world

How to test drive a car | Cars
www.telegraph.co.uk/cars/advice/...ve-a-car Words: function object return var window
Cached Similar Inlinks IP:96.17.148.41 Score:7.80

Fig. 6 An Advertisement on Search Results Page

VI. Comparisons with Existing Auction and Ad Systems

We would now like to compare our system to existing auction and ad systems. We first note that our daily auction mechanism does not depend on the fact that we are selling ads versus some other product, nor does it depend on the ads themselves. Further, it does not depend on any property of the bidders beyond their bids. So our system meets the definition of an auction as given in Section IV. It should be noted that systems which estimate click-through rates of a particular ad probably fail the first criteria, so it questionable whether they are truly auctions unless one views that as part of the bid. Our auction system is open in that an advertiser knows the totals of previous bids for the keywords in the date range they are trying to place their ad. Our minimum bid requirement ensures also that the auction is ascending price. Unlike AdWords and Bing Ads, our system currently only places one ad on a search result page, so it is perhaps most natural to compare our system with a usual open ascending price auction or equivalently a sealed, second price auction system. First, though, we would like to compare our system to lotteries and all-pay auctions with which our system at least bears some similarities.

As presented in the last section, the selection of which ad to display make use of a pseudo-random number generator which may seem somewhat lottery-like in nature. As many places have legal requirements on lotteries, it is interesting to compare our system to a lottery. Here we are more interested in various legal definitions of lotteries than in the game theoretic definition of a lottery as a probability distribution over a set of prizes. As this is not a law paper though, we will only consider U.S. Code. In U.S. Code, a lottery is defined in 12 U.S.C. §25c as:

> Any arrangement, other than a savings promotion raffle, whereby three or more persons (the participants) advance money or credit to another in exchange for the possibility or expectation that one or more but not all of the participants (the winners) will receive by reason of their advances more than the amounts they have advanced, the identity of the winners being determined by any means which includes – (A) a random selection; (B) a game, race, or contest; or (C) any record or tabulation of the result of one or more events in which any participant has no interest except for its bearing upon the possibility that he may become a winner.

In our system, an ad view is not easily exchangeable into an amount of money, so to some degree this portion of the definition a lottery is not met by our system. Further, all accepted advertisers, not just some advertisers, will receive close to their share of the search views for that keyword for that day provided that the pseudo-random number generator (in our case, Mersenne Twister) iterates through the sample space reasonably uniformly and the number of searches on the keyword for a given day is large enough. A pseudo-random number generator is completely deterministic and is only used for splitting the search views into shares. The choice of generator and starting seed could be published without affecting much how bidding would go in our system provided the number of searches on a keyword was large enough. On the other hand, if one could calculate in advance whether or not a particular lottery ticket's number is going to be chosen, it would likely effect whether or not one would purchase that ticket. So the (A) criteria of the winning criteria of the definition above fails. Unlike (A) where the winning event is chosen by a random process, winning events of type (B) and (C) deal with events which are chosen from a sample according to some unknown distribution. For example, in betting on a horse race, if one completely knew all the details of the conditions input to the race, one might be able to predict the output, but this is not what one typically knows – one typically does not even know the distribution of those inputs. In our system, though, those inputs and mechanism can be known, and it does not greatly effect the bidding.

An **all-pay auction** is one in which all bidders must pay the value of their bids, but only the highest bidder wins the auction. This is similar to our auction in that all bidders in our system must pay; however, in our system the amount bid determines the fraction of the day's views (the prize) one receives. Forms of all-pay auctions are used in penny auction sites like Quibids.com, Beezid.com and others. The two-person, dollar auction variant of an all-pay auction has been used to model conflict escalation [11]. All-pay auctions are also used as models for political lobbying, for job-promotion competitions, and for research and development competitions. Che and Gale [1] have shown that all-pay auctions generate higher expected seller revenues than first price, sealed-bid auctions, and hence, also second price, sealed-bid auctions which corresponds to ascending price auctions. I.e., all things being equal, one would expect a research contest among many applicants with a prize to generate more research effort than a single best researcher who was awarded a grant for the same amount chosen from a pool of grant applicants. One can construct situations where a property like this holds for our ad auction system versus an open auction system. For example, consider a two bidder situation for a keyword which is gets 100 impressions/day. Suppose both bidders value an impression at 1 ad credit (CR), but both of whom have a limited budget of 50CR. In the open ascending price auction system, the first bidder would bid 50CR. The second bidder might want to bid higher, but cannot because of his limited budget. The first bidder gets all the impressions for 0.5CR each. On the other hand, the auction house has lost potential revenue, and the second bidder does not get their ad displayed at all. Our auction system allows the second bidder to bid 50CR as well. So each bidder is expected to get half of the 100 impressions, that is, 50 each, and they would each be paying 1CR/impression. So in this situation our ad system generates more seller revenue than an open ascending price auction. Recall that seller revenue was one of the metrics we mentioned Section IV used to evaluate auctions. Our system also gives more bidders an opportunity to have their ads displayed at a fair price increasing total bidder payoff, one of the other auction system evaluation metrics mentioned in Section IV.

The third metric we presented to judge an auction system was truthfulness in bidding. Notice in our system that if an advertiser's valuation of a keyword on a given day is more than twice that of any other advertiser, then if the advertiser bids their valuation, no other advertiser will try to bid as half the total bid after the first bid would be more than their valuation. So in this situation truthful bidding would prevent having to share keyword impressions for a given day. The situation where advertisers have the same valuation for keyword also promotes truthful bidding. To see this we show that our system behaves like a usual open ascending price auction in this case. Suppose for the keyword lemonade searches generate on average 100 impressions in a day. If two advertisers both value an impression at 1CR, then both would value the lemonade keyword for one day at 100CR. Consider a usual auction. The first advertiser would bid 100CR, and the second advertiser would not bid as the cost would be more than a 1CR/impression. The first advertiser would not underbid as then the second one would have the opportunity to take all 100 impressions. Now consider what would happen with Yioop's bidding system. If the first person bids 100CR for lemonade for the day, then the second person would have a minimum bid of 50CR. If the second person bids 50CR, then they will receive about $50 \cdot 100 + 50 \cdot 100 \approx 33$ impressions, so the cost/impression will be over a credit. Higher bids would only make the cost/impression worse, so the second bidder would not bid. Suppose the first person underbids some value $1 \leq x < 100$, then the second bidder can truthfully bid $100 - x$, and the cost/impression still works out to 1CR/impression for both parties.

VII. Experiments

After implementing our advertising system in Yioop, we performed some experiments to check its design and functionality. We did two types of experiments, load testing and user acceptance testing.

For load testing, we checked server response times when multiple users accessed the system at the same time. Load testing was performed on single machine. We did tests involving measuring response time while varying the number of simultaneous requests using JMeter [18] and Apache Bench. We did two experiments. In our first experiment to make a request, we randomly chose between nine queries phrases. We then compared the response times if our ad system was disabled versus if each query had one advertisement that could be displayed. For each response time data point, we did the same test three times and averaged the results. In our second experiment, we will still randomly chose between nine different queries, but we also varied the number of ads that could be displayed. I.e., we ran the test where each query might have a choice between 5 ads, 10 ads, or 20 ads. Again, for each response time data point, we did the same test three times and averaged the results. When choosing bid amounts for our ads, we always chose the minimum possible bid.

Our user acceptance tests involved getting users to use our ad system. The aim of these tests was to check whether users can use the newly designed advertisement platform easily or not. As users carried out tasks given to them and experimented with our system, we noticed difficulties they faced. This in turn led to us modifying the user interface, help text, and simplify the ad work flow.

1) **Load Testing**

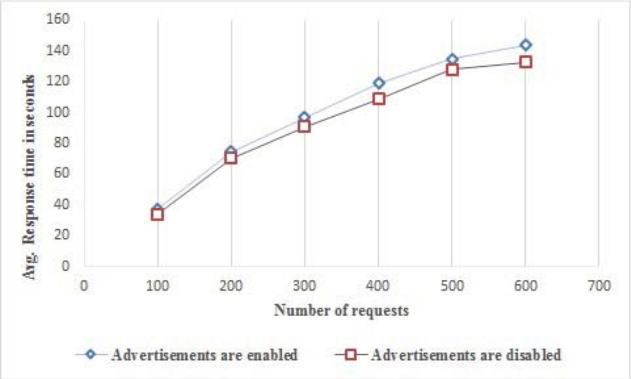


Fig. 7 The Number of Concurrent Requests versus Response Time for Single versus No-Ad Cases

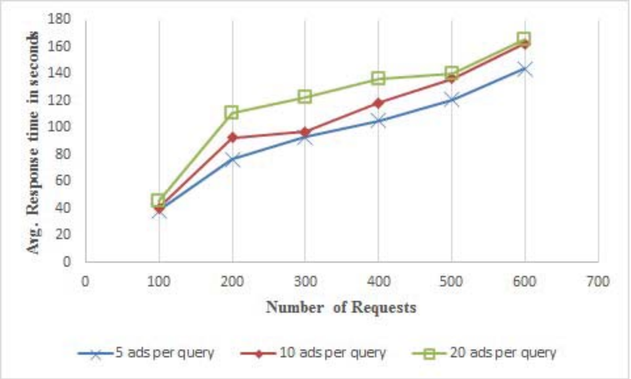


Fig. 8 The Number of Concurrent Requests versus Response Time While Varying the Number of Ads

From Fig. 7, one can conclude that the response times when advertisements are enabled versus disabled are fairly close. This can be quantified a little bit more by making a table of the percent difference in response times. We have done this in Table VI which shows that the overhead in incorporating ads into the existing Yioop search engine is minimal.

Fig. 8 considers the case where there was a choice between the ad to display. In this case, there are many advertisement related to the entered query and the auction system has to come up with single choice. As one can see, the graphs are fairly close even when the system had to choose from about 20 ads.

TABLE VI
PERCENT CHANGE IN RESPONSE TIMES WHEN ADVERTISEMENTS ARE ENABLED

Number of users	Percentage increase in response time
100	3.11
200	2.05
300	2.07
400	2.57
500	1.40
600	1.89

2) **Usability Testing**

Nielsen [12] suggests that the maximum benefit from website usability studies is obtained by performing tests with 3-5 users and using the results to iterate site design. Nielsen [13] suggests five quality components to consider: learnability, efficiency, memorability, errors, and satisfaction. After finishing our first ad system prototype, we conducted usability tests on three users. We were both interested in the usability of the ad purchase interface and in the visibility and usability of displayed advertisements. Users were given two tasks: To create an advertisement with a given set of keywords and to find their ad in search results after the advertisement was purchased. Efficiency was measured in terms of time to perform these tasks and averaged three minutes for the former and a minute or so for the former. This was for the first time performing these tasks. In the initial design there was some confusion about how to enter keywords and in the labeling of where to enter the url the ad is supposed to take user's to. To increase user satisfaction we created an ad preview, added help text for the keyword area, and redesigned some of the buttons and label text.

VIII. CONCLUSION

We have described our online advertisement system, properties of its ad selling mechanism, and its implementation extending the Yioop search engine. As we have shown, our ad selling mechanism is an auction which promotes truthfulness in bidding. We have argued that our mechanism can in some circumstances yield higher seller revenue and total bidder payoffs than a traditional open ascending price auction/sealed second-price auction. We described our load testing results of our system which indicate it is robust in reasonably large user contexts, and we also described our usability experiments and the improvements we made to our system as a result.

We envision several future improvements and extensions to our system. As the Yioop search platform is designed to be run by small to medium scale operators, the ad system may end up being used on lower trafficked sites which makes it harder to sell keyword advertising. On the other hand, if there was a common distributed network where ad buyers could buy ads which could appear on any Yioop instance, there might be a bigger market. So one extension might be to implement such a distributed ad network. Other extensions might be to vary the rules by which ads are displayed. For example, one might add a rule, that if there were too few searches on a keyword for the day, then some kind of partial refund of ad credits could be given. At this point although our system does support view and click analytics for prior campaigns, this information is not used in our system to forecast the likely success of future campaigns, and this could be a useful enhancement. The mechanism we have described in this paper only places at most one ad on a search result page. Systems like AdWords and Bing Ads allow for the placement of multiple ads through the use of the general second price auction mechanism. It would be useful to extend our system so as to allow the placement of multiple ads on a search results page.

As our system seems like a promising alternative to current ad selling systems, we expect to continue to make improvements to it.

ACKNOWLEDGMENTS

The authors would like to thank Paul Thienprasit for helpful discussion during the writing of this paper.

REFERENCES

[1] Y. Che and I. Gale. *Expected Revenue of All-Pay and First-Price Sealed-Bid Auctions with Budget Constraints*. Economics Letters. 1996. Vol. 50. pp. 373–379.

[2] Mung Chiang. Networked Life: 20 Questions and Answers. Cambridge University Press, New York. 2012.

[3] Dino Grandoni. *AOL in Deal With Microsoft to Take Over Display Ad Business*. Technology Section. New York Times. June 29, 2015.

[4] C. Karande, A. Mehta, R. Srikant. *Optimizing Budget Constrained Spend in Search Advertising*. Sixth ACM International Conference on Web Search and Data Mining, WSDM 2013, ACM, pp. 697–706.

[5] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. *AdWords and Generalized Online Matching*. Journal of the ACM. Volume 54, Number 5. 2007.

[6] ComScore Inc (2011) "The 2010 U.S. Digital Year in Review" (Press Release) Retrieved on May 1, 2015 from, http://www.comscore.com/Insights/Press-Releases/2011/2/comScore-Releases-The-2010-U.S.-Digital-Year-in-Review

[7] Google Inc. Google AdWords. Retrieved on Dec. 29, 2015 from, http://www.google.com/about/company/history/.

[8] Google Inc. Google AdWords. Retrieved on May 1, 2015 from, https://www.google.com/adwords/

[9] V. Krishna. Auction Theory, 2nd Ed. Academic Press. 2010.

[10] Donald R. Lehmann and Charles B. Weinberg. *Sales through Sequential Distribution Channels: An Application to Movies and Videos*. Journal of Marketing. Vol. 64, No. 3 (Jul., 2000), pp. 18–33.

[11] B. O'Neill. *International escalation and the dollar auction*. Journal of Conflict Resolution. Vol 30. pp. 33–50.

[12] J. Nielsen. *Why You Only Need to Test with 5 Users*. March 19, 2000. Retrieved on Dec 27, 2015 from https://www.nngroup.com/articles/how-many-test-users/.

[13] J. Nielsen. *Usability 101: Introduction to Usability*. January 4, 2012. Retrieved on Dec 27, 2015 from https://www.nngroup.com/articles/usability-101-introduction-to-usability/

[14] Sagin, Erin (January 22, 2014).“All About AdChoices: What is AdChoices? How Does it Affect Advertisers and Consumers?” WordStream. Retrieved on May 1, 2015

[15] Bing Ads Help. Microsoft. Retrieved on Dec. 28, 2015 from http://help.bingads.microsoft.com/

[16] Google Inc. Inside AdSense. Retrieved on Apr 30, 2015 from, http://adsense.blogspot.com/

[17] Bradley Johnson (December 8, 2013). 10 Things You Should Know About Global Ad Market. Retrieved on May 1, 2015 from, http://adage.com/article/digital/online-measurement-16-web-clicking-display-ads/139367/

[18] Apache JMeter Retrieved on Apr 30, 2015 from, http://jmeter.apache.org/usermanual/index.html

[19] Seekquarry/Yioop About Page. Retrieved on Apr 30, 2015, from, http://www.seekquarry.com/p/About