

# Maya : Zero Effort Personalized Web Experience for the Entire Internet User-base

Leo Arulraj ( Email: arulraj@cs.wisc.edu )

*Department of Computer Sciences, University of Wisconsin, Madison*

## Abstract

*A highly personalized version of the vast, ever-growing World Wide Web and related services that contains only information that is interesting and useful to a common Internet user will benefit him/her a lot. The exponential growth trends in the number of web sites over the past decade is clear proof of this information overload problem. Such a personalized web will try to bring interesting content to the user rather than the user searching for content on the web. However, privacy concerns have made it impossible to collect complete web usage habits of every Internet user. This has resulted in ineffective personalization solutions so far which have one or more shortcomings : they target only a very small fraction of the Internet user-base ( e.g. users of slashdot.com, reddit.com etc ) , require non-trivial effort from the user in order to send feedback votes of 'like' / 'dislike' for every web page visited back to the personalization service, are imprecise due to lack of complete web usage information of all Internet users , they are not holistic and only personalize specific web services ( e.g. Google Search ). This paper describes Maya a solution that attempts to address all the above shortcomings and provide a highly personalized web experience for everyone. Web usage traces of the entire Internet user-base is collected anonymously and aggregated in a peer to peer fashion. The collected trace is processed using machine learning algorithms and the processed information is sent to a browser plugin that personalizes the web for the user. Maya can greatly enhance the quality of several Internet services like Online Advertising, Web Search and Internet Content Discovery.*

## 1 Introduction

“The Internet is really about highly specialized information, highly specialized targeting.” *Eric Schmidt, CEO, Google.*

Since its inception in the 1960s, the Internet has grown enormously in several respects. As of 2010, the total number of hostnames on the Internet is about 256 million out of which 77 million are active and the total number of Internet users are 2 billion [3]. The common Internet user in the United States spends about 2 hours online a day and visits about 80 non-unique web pages a day [3]. These

facts hint at the huge amount of information available on the Internet and the slow rate at which a user consumes information from the Internet. A highly personalized version of the Internet that tries to help a user find interesting resources on the web will result in a productive web usage.

If complete information about how every user browses the web is available, then effective personalization is possible. Since Internet users are concerned about their online privacy, they are unwilling to completely trust a third party service with their complete browsing history. Several approaches have been suggested to resolve this *privacy-personalization conflict* but all of them have shortcomings. §2 presents a survey of existing approaches along with observations that try to convince the reader that web personalization is not completely solved. In particular, the advertisements on the Web today can be made more relevant and also interesting web pages can be recommended to users providing a rich web usage experience.

One approach that is widely used in practice is to allow the users to create user accounts with a service that attempts to provide a personalized web experience ( e.g. StumbleUpon.com ) and then voluntarily vote 'like' or 'dislike' on websites visited. However, this method of voluntary sharing of browsing information along with user identity has not been effective in practice because of two significant issues. The first issue is that only a very small fractional subset of Internet users are members of such services. The second issue is that users with such accounts do not provide their votes for all the websites they visited either due to privacy concerns or due to the effort involved in doing this all the time.

Huge scales of browsing traces collected from everyone on the Internet will tremendously help in effective personalization. A good solution is to collect browsing traces of the entire Internet user-base anonymously and automatically with zero effort from users respecting their privacy concerns. Every user's browsing trace is broken down into sets of related web sites and then sent to an aggregation service anonymously. Such sets of related web pages from every user in the Internet are aggregated and analyzed for patterns. The scale of the collected information will re-

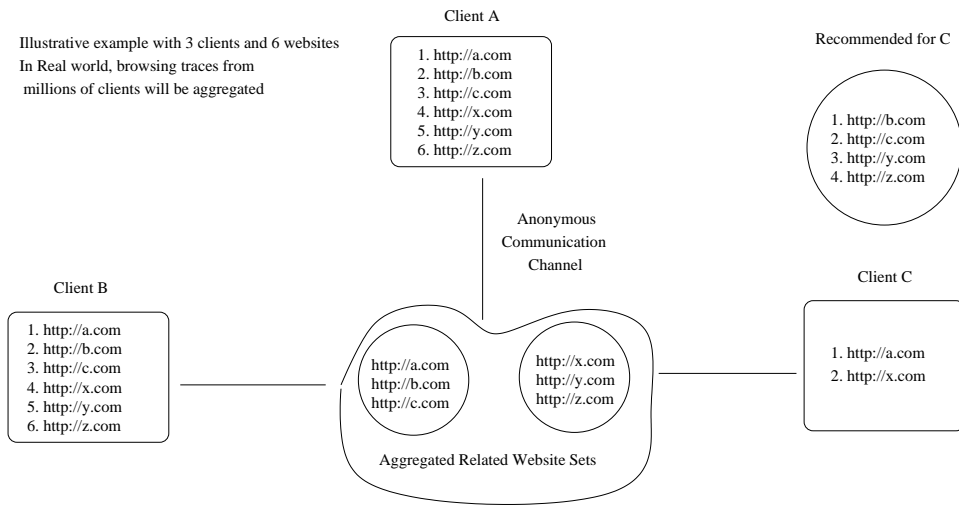


Figure 1: **Illustrative example with 3 Clients.** Clients A and B send their related website sets which is aggregated by the peer to peer aggregation infrastructure. Client C's web experience is personalized using the aggregate information and the websites visited in the past. It should be noted that the real scale of aggregation will consist of approximately 2 billion users.

veal useful browsing patterns. A mechanism for personalizing any service related to the Internet ( e.g. advertising, recommendations ) without constraining to specific services from specific brands can be built using the traces collected. The anonymous nature of the aggregated information preserves the privacy of the users. Automatic collection of information requires absolutely zero effort from the user towards enhancing personalization. Collection of browsing traces at this scale ( from all Internet users ) will help in finding browsing patterns and help effective personalization.

This paper describes *Maya*, a solution that uses the above ideas. At a high level, it anonymously aggregates the browsing information of the entire Internet user base using anonymous overlay networks and peer to peer solutions. The anonymized aggregate information is processed using machine learning techniques to infer useful browsing patterns and also to classify related websites into groups ( e.g. web pages on Philosophy ). This processed information is combined with the complete browsing history of an individual user in order to personalize Internet related services for that particular user. Machine learning techniques are heavily used for both personalization and to handle maliciousness. A simple illustrative example of how *Maya* works is shown in Figure 1. The design of *Maya* with high level directions, challenges and scope for further Research is presented in §3.

*Maya* has the potential to improve the quality of several Internet related services. For example, the multi billion dollar Online Advertising industry is directly affected by the "privacy-personalization conflict" [12]. *Maya* can enable targeted advertising without infringing on the privacy of the Internet users. *Maya* can also improve the quality of existing personalized web search, web content

discovery tools, e-recommendations.

## 2 Existing Services, Techniques and their Limitations

With the aim of convincing the reader that web personalization is an unsolved problem, this section lists down all existing web personalization services and techniques suggested by previous research for achieving web personalization along with the shortcomings.

### 2.1 Personalization Services in Reality

StumbleUpon [5] is a popular web personalization service that combines collaborative human opinions with machine learning of personal preference to create virtual communities of like-minded web surfers. It generates web site recommendations based on the ones that a user has visited and/or liked. Users create accounts and then provide information on whether they likes/dislike websites and optionally tag the website to a category. A recommendation engine uses classification and clustering techniques on the collected human opinions to come up with recommendation of new websites to users. SimilarWeb [5] is a service that provides a web surfer with a list of websites that are similar to the one he is currently browsing. SimilarWeb ( part of SimilarGroup ) runs web crawlers called Similarity Engines that analyze the properties of web pages like keywords, hyperlink structure and automatically classify and tag them into fine grained categories. Recently, social networking based approaches for collecting collaborative opinions of like minded users have become popular too ( e.g. recommendations plugin from Facebook.com [5] ).

Social news and website aggregators ( like Reddit.com, Digg.com, Del.icio.us ) allow its users to post links to popular content on the web under major topics. Web surfers can visit the topic pages of their interest and get access to a list of relevant popular websites. These services are purely driven by their user base. There are a host of similar services on the Internet.

All of the above techniques have the following shortcomings : a) Users need to put in non-trivial effort to constantly vote 'like' or 'dislike' on all the websites they visit in order to get good personalization, b) Users need to trust a third party service with their personal browsing traces, c) These services do not have a holistic model of the general tastes of the user because they operate on partial traces, d) These services are used by fractions of the entire Internet user base ( for example, as of December 2010, StumbleUpon has a total of 13 million members which is barely 0.007% of the total number of Internet users). e) Some of these techniques apply only to specific parts of the web ( for eg. Facebook recommendations plugin is applicable only to websites that use this plugin, Google Search personalization is application only to users who search the web through it. )

## 2.2 Other Techniques

Secure Multi-Party Computation ( SMC ) [7] schemes allow multiple parties holding secrets to know the aggregate of all their private secrets without revealing one individual party's secret to any other party involved in the aggregate computation task. SMC has been applied to the general field of collaborative filtering for recommendation and implementation techniques have also been proposed. Since Internet users need privacy of their browsing habits and not secrecy SMC based approaches are an overkill.

Client Programs transmitting sets of related websites in *Maya* can do so under an assumed pseudonym in order to maintain privacy. [9, 10]. However, there is a high risk of losing the privacy once an adversary is able to relate the pseudonym with the real identity of the client.

## 3 Design of *Maya*

*Maya* consists of two significant components : 1) The Anonymous Aggregation Infrastructure ( *A2I* ) §3.1 used to aggregate the complete browsing history of the entire Internet user base. 2) The Decoupled Privacy Preserving Personalizer ( *DP3* ) §3.2 specific to each individual user that personalizes the Internet and Web services using the complete non-anonymous knowledge of her/his browsing habits.

### 3.1 *A2I*: Anonymous Aggregation Infrastructure

The aggregation is done using a peer to peer system consisting of a software component called *A2I node* on every computer. Each *A2I node* plays several roles: 1) *Transmitter* 2) *Router* 3) *Aggregator* 4) *Filter* 5) *Manager* 6) *HoneyPot*. Detailed discussion of the various functions of an *A2I node* follow.

#### 3.1.1 *Transmitters and Routers*

During an Aggregation task, every *Transmitter* anonymously sends the browsing history information in the form of *sets of related websites* called *RSet* to an assigned aggregator. The *Transmitter* is trusted not to cause intentional risk to the privacy of the user because it is part of the common software package downloaded and installed by every user who wants to use *Maya*. The browsing trace is broken down into *RSets* of size two or more before transmission in order to reduce the risk to the users privacy from adversaries who might try to resolve the trace back to the client. For example, suppose an *RSet* contains 1000 related web sites. If the *Transmitter* were to send these 1000 web sites as one *RSet*, a malicious *A2I node* cooperating with an adversary that has access logs from malicious websites hosting a subset of 50 web sites can try to resolve the identity of the user who sent the *RSet*. Then the adversary will get to know the other 950 websites that the same user visited. Adversaries can also get access logs from malicious ISPs.

One way to tackle this is for *Transmitters* to create *RSets* smartly using the knowledge from past Aggregated information. If a particular set of websites was already sent in the form of *RSets* several times in the past, then the *Transmitter* can safely send a new *RSet* with those web sites because the adversary can only figure out that an *RSet* can belong to one of several real identities who all have also visited the same set of websites. The adversary cannot say with confidence that it is one particular user who sent the *RSet* in question. This is also called K-Anonymity [11] and therefore cannot zero in on one particular real identity.

Malicious *Transmitters* can transmit a fake *RSet* to confuse *Maya* and benefit thereby. For example, an adversary can use malicious *A2I nodes* to send *RSets* that try to make a commercial website related to several other web sites. This way there will be more traffic and more sales for the commercial website. *Maya* handles such fake *RSets* by using *Filters*. *Filter* nodes present alongside the *Routers* detect lies by malicious *Transmitters* and discard them away to prevent them from being sent to an *Aggregator*. Simple keyword based techniques can be used to detect if two given websites are related or not. ( e.g. two related websites might contain keywords from same top-

ics like Philosophy or Religion while unrelated website wont.) More sophisticated machine learning algorithms can also be used. Once a lie is detected, it is also possible to feed this information to the *Routers* to block or throttle further transmission from the malicious *Transmitters*.

The *Routers* form the building blocks of the anonymous overlay network similar to the popular TOR onion routing that serves as the communication channel between *Transmitters* and *Aggregators*. Malicious *Routers* can drop or incorrectly route *Rsets* and thereby hindering progress of an aggregation task. One possible approach to handle malicious *Routers* is to set up fresh Routing tables in the Anonymous Overlay Network for each aggregation task and using peer to peer monitoring to detect, control and punish maliciousness during Routing.

### 3.1.2 Aggregators and Filters

In every Aggregation task, the *Aggregators* form a logical tree hierarchy amongst themselves with *Transmitters* as leaf nodes. Every *Aggregator* collects information from its children. The *Aggregators* invoke the *Filters* on the *Rsets* that they received before sending it to their parent *Aggregators*. This approach detects and controls the effects of the malicious *Transmitters* and malicious *Aggregators*. Once detected the trust scores associated with the secure pseudonyms of the corresponding *A2I node* can be suitably adjusted to give them lesser preference in future aggregation rounds. *Aggregators* with higher trust scores are placed higher up in this logical hierarchy by the *Managers* during the creation of an Aggregation task. *Aggregators* also run statistical analysis on the results of an aggregation task and compare it with aggregate information from previous aggregation tasks to further filter out the possible effects of malicious *A2I nodes* that escaped through the *Filters*.

Extensive previous research on Web Personalization [1] can be leveraged to aid in using the raw aggregated information collected to provide useful personalization services. In fact, annual workshops have been conducted on Intelligent techniques for web personalization for the past 10 years [2]. The aggregate information can be processed into the following useful information : a) "sets of related websites" called *Golden Sets* b) information on intersection , overlap, partial and full containment amongst several *Golden Sets* c) tags and keywords that classify the *Golden Sets* generated using machine learning techniques d) information on which websites are related to commercial websites for targeted advertisement. Additional information can also be generated when needed for newer Applications of *Maya*.

The *Filters* use a variety of sophisticated machine learning techniques to thwart the effects of malicious *Aggregator* and *Transmitter A2I nodes*. *Filters* can leverage the enormous amounts of research and implementation

on using Artificial Intelligence and Machine Learning for Web Personalization [2, 1, 5] to detect and ignore *Rsets* that contain unrelated websites. Once malicious nodes are identified, information can be sent to the Anonymous Routing infrastructure and other *A2I nodes* to block or control the further damage that such nodes can cause.

### 3.1.3 Manager

*Managers* perform tasks like : creating and conducting aggregation tasks, the *A2I nodes* that should be part of a given aggregation task, the roles that different *A2I nodes* should take during an aggregation task, how many repetitions of an aggregation task should be done to tackle the effects malicious *A2I nodes* using statistical techniques, maintain and update trust scores for each pseudonymous *A2I node* to aid in role assignment, etc. The policy used by *Managers* while assigning roles to *A2I nodes* can be based on the several properties of an *A2I nodes*. Each *A2I node* has a trust score assigned to it and maintained over time that correlates to factors like non-maliciousness while performing function other than Transmitting. These trust scores can be used by *Managers* while assigning roles to *A2I nodes*. Other factors like availability of an *A2I node* for participation in aggregation tasks over time, contribution of resources by an *A2I node* in the past aggregation jobs, geographical proximity to other *A2I nodes*, etc can also be used. Malicious *Managers* can be handle by using a quorum of *Managers* for making decisions and thereby making it possible to find and punish maliciousness.

### 3.1.4 HoneyPot

The aggregate of all aggregation tasks so far is called *Golden Aggregate*. *HoneyPots* store and server : a) The *Golden Aggregate* b) Snapshots of *Golden Aggregate* over time c) the aggregate from individual aggregation tasks *HoneyPots* serve processed aggregate information to the *DP3* component and to other web services to achieve precise personalization. *HoneyPots* also serve information to *Aggregators* to help with statistical analysis and pruning of malicious information from individual aggregation tasks. *HoneyPots* can also serve the information they hold to web services like Search Engines tools to enhance their quality of service rather than personalizing them ( for example, Search Engines can use the aggregate information from *HoneyPots* to rank search results).

Malicious *HoneyPots* can serve incorrect information to a requesting *DP3*. To handle this, checksums of processed aggregated information can be generated and independently served by other *HoneyPots* to aid detection of maliciousness. *DP3* can also send multiple requests to several *HoneyPots* and thereby detect malicious *HoneyPots*. Once detected malicious *HoneyPots* can be reported to the *Managers*.

### 3.2 DP3: Decoupled Privacy Preserving Personalizer

The *DP3* component can be implemented as another function within an *A2I node* or as a heavy weight web browser plugin or a combination of both. *DP3* maintains detailed information on which web sites were visited by the user at what times in a secure fashion.

It can use sophisticated machine learning techniques to build behavioral models and taste models of the user over time. Enormous amounts of previous research and implementation on Artificial Intelligence for Web Personalization can be leveraged [2, 1, 5]. *DP3* obtains aggregate information collected about the websites related to the user's browsing model from the *HoneyPots*.

*DP3* can use smart techniques to make sure it does not reveal aspects of the user's browsing model and preserves the privacy of the user when it requests for information from *HoneyPots*. Two preliminary directions to achieve this would be : a) to ask for "more information than needed" and to ask for "information that is not needed" b) to spread requests across several *HoneyPots* that are less likely to collude together.

The precise *Taste Models* of a user can be translated into compact strings called *Taste Strings* and then set as cookies while accessing web sites. This enables the Web Services to understand the tastes of the user and target the service to the user. It should be noted that the taste information has been voluntarily shared by the user with the web service and the web service did not get this by tracking the user using third party cookies or in any other way invading his privacy.

Compaction of the *highly informative Taste Model of a user* into a *compact Taste String* which can be stored in a couple of Bytes is an exciting research problem. One possible approach is to hash the identifiers of *Golden Sets* that are related to the user from the *HoneyPots* and then set bits in a string corresponding to the hash values similar to Bloom Filters [4]. This string has *Golden Set* containment information and can be used as a *Taste String*. The web service can use the identifiers of the *Golden Sets* obtained from *HoneyPots* to decipher the taste of the user from the *Taste String*. Many of the Internet related services like Web Search, Online Advertising, Content Discovery can be effectively personalized using *Taste Models* and *Taste Strings*.

## 4 Conclusion

Web Personalization is a pressing but unsolved problem. A survey of existing solutions and techniques proposed by previous research are presented and convincing arguments are made to show that all of them have significant shortcomings. A novel solution to this problem is outlined with

high level design directions along with challenges and research problems.

Challenges in building *Maya* and scope for research arise in : controlling the effects of Maliciousness using machine learning and statistics, providing accountability for the actions of various functions and building trust models that ensure correct operation of *Maya* when running on untrusted computers on the Internet ( Techniques like PeerReview [8] can be used to achieve accountability ), efficient conduction of aggregation tasks by minimizing the network and computing resources, scaling *Maya* to the size of the Internet, storing and distribution of the aggregated information optimally, etc.

The initial plan is to build a research prototype of *Maya* to demonstrate its usefulness using a simulated user base ( with browsing habits influenced by manually categorized pages from Internet directory [6] ) and simulated malicious nodes.

## References

- [1] *The Adaptive Web, Methods and Strategies of Web Personalization*, volume 4321 of *Lecture Notes in Computer Science*. Springer, 2007.
- [2] Workshop on intelligent techniques for web personalization, yearly. <http://ls13-www.cs.uni-dortmund.de/homepage/itwp2011/index.shtml>.
- [3] NetCraft Ltd. Dec 2010 Web Server Survey, World Internet Users and Population Stats, Neilsen Average U.S. Internet Usage Survey. Websites, 2010. <http://news.netcraft.com/archives/2010/12/01/december-2010-web-server-survey.html>, <http://www.internetworldstats.com/stats.htm>, [http://blog.nielsen.com/nielsenwire/online\\_mobile/june-2010-top-online-sites-and-brands-in-the-u-s/print/](http://blog.nielsen.com/nielsenwire/online_mobile/june-2010-top-online-sites-and-brands-in-the-u-s/print/).
- [4] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13:422–426, July 1970.
- [5] Stumble Upon - Discover the Best of the Web, SimilarWeb - The Ultimate Discovery Tool, Facebook Recommendation Plugin. Websites, 2010. <http://www.stumbleupon.com/>, <http://www.similargroup.com/our-technology.php>, <http://developers.facebook.com/docs/reference/plugins/recommendations>.
- [6] List of web directories. Website, June 2010. [http://en.wikipedia.org/wiki/List\\_of\\_web\\_directories](http://en.wikipedia.org/wiki/List_of_web_directories).
- [7] O. Goldreich. Secure multi-party computation. Working Draft, Mar. 2000.
- [8] A. Haerberlen, P. Kouznetsov, and P. Druschel. Peerreview: Practical accountability for distributed systems. In *Proceedings of the 21st ACM Symposium on Operating Systems Principles (21st SOSP'07)*, Stevenson, Washington, USA, Oct. 2007. ACM SIGOPS.
- [9] Kobsa and Schreck. Privacy through pseudonymity in user-adaptive systems. *ACMTIT: ACM Transactions on Internet Technology*, 3, 2003.
- [10] A. Kobsa. Privacy-enhanced web personalization. In *The Adaptive Web, Methods and Strategies of Web Personalization*, volume 4321. Springer, 2007.
- [11] L. Sweeney. k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10:557–570, October 2002.
- [12] Network Advertising Initiative. Website, 2010. <http://www.networkadvertising.org/>.