

Automatic profile generation in eRACE

Christiana Christophi
Dept. of Computer Science
University of Cyprus
PO Box 537, 1678 Nicosia
CYPRUS
cs98cc1@ucy.ac.cy

Marios Dikaiakos
Dept. of Computer Science
University of Cyprus
PO Box 537, 1678 Nicosia
CYPRUS
mdd@ucy.ac.cy

ABSTRACT

In this paper, we describe the design of a profile generator toolkit, which aims to automatically create realistic user profiles for a mobile personalized portal service. These profiles simulate actual user behavior and are utilized to experiment with a personalized crawling system, called eRACE. With the help of this tool, we will assess the performance and scalability of the eRACE system by creating appropriate models.

Categories and Subject Descriptors

H.3.5 [Online Information Services]: Web-based services – XML, portal, profile registration.

General Terms

Measurement, Performance, Design, Experimentation.

Keywords

User profile, XML, search engine, performance, testing.

1. INTRODUCTION

The *extensible Retrieval, Annotation and Caching Engine* (eRACE) [4], is a modular, programmable and distributed proxy infrastructure that collects information from heterogeneous Internet sources according to XML-encoded *eRACE profiles*. Collected information is stored in a software cache for further processing, personalized dissemination to subscribed users, and wide-area dissemination on the wireline or wireless Internet. eRACE supports content-assembly, customization, personalization, service differentiation, request scheduling and resource optimization by enabling the registration, maintenance and management of profiles representing the interests of individual users or the definition of portal services.

In order to test the performance and scalability of the eRACE infrastructure, we developed a toolkit that aims to simulate an environment within which the eRACE system will be tested. This toolkit generates automatically eRACE profiles that reproduce user interests. The profile generator employs statistical models of WWW use and input parameters to create profiles that can drive the eRACE infrastructure under realistic conditions. In this paper, we describe the profiles defined and used in eRACE and the design of the profile toolkit.

Information regarding user interests, priorities, filtering directives, service levels, connection modalities, etc. is maintained in eRACE in terms of XML-encoded profiles that are distinguished

into two types: the semantic and the execution profiles.

2. PROFILES IN eRACE

2.1 Semantic Profiles

A semantic profile encodes a generic portal structure, which can be easily customized to represent different portals, for instance personalized hierarchical collections of information, (e.g. Yahoo!). It consists of a list of topics organized in a tree-like arrangement. Each topic consists of either source or topic. Through a single interface, a user or an administrator can easily configure the generic structure according to personal preferences by declaring topics, pinpointing and prioritizing information sources of interest and setting filtering directives (e.g. keywords). Moreover, the user can set presentation preferences that determine the placement and appearance of information.

2.2 Execution Profiles

An execution profile is partly derived from the corresponding semantic profile. It represents the “processing” requirements that the semantic profile places upon eRACE. It contains directives to the system regarding the information to be collected on a user’s behalf and the subsequent processing required to deliver the information in the appropriate form. Furthermore, an execution profile specifies the frequency of its execution; for example, how often pages are fetched on a user’s behalf, according to application-level QoS policies. In a nutshell, execution profiles represent the workload that drives eRACE; they enable the creation of models that predict changes in resource requirements with a varying user-base, support dynamic capacity planning and better scheduling of operation. Furthermore, it is possible to implement QoS policies linked to pricing and resource consumption cost.

3. PROFILES GENERATION TOOLKIT

In order to analyze eRACE performance and scalability, we need to test it under realistic workloads. Therefore, we need to automatically produce synthetic collections of *execution* profiles representing a varying volume of realistic user requests. These execution profiles must be generated out of semantic profiles bearing characteristics of real user interests.

3.1 Semantic Profiles Generation

Semantic profiles consist of a collection of sources, classified by topic. The number of *sources* contained in a profile, combined with the crawling *depth* of each source, determines the workload incurred by eRACE for that particular profile. Moreover, *keywords* and *keyword-weight* drive the post-crawl processing; keywords belong to a topic and specify filtering actions upon a collected source, whereas the weight identifies the importance of a keyword.

The toolkit receives five inputs: (1) the number of users; (2) a seed file of URLs classified by topic; (3) a list of keywords classified by topic; (4) probabilities for selection of the *depth* attribute and (5) probabilities for selection of the *weight* attribute for each keyword. The seed file is actually a large list of URLs based on the taxonomy of Open Directory Project (ODP) [1]. The ODP provides a large, human-edited Web directory available in RDF encoding. We rank the sources by popularity according to the backlinks count obtained from Google [2]. Intuitively a page referenced by many is more important than a rarely referenced one. Thus, we use a collection of actual URLs organized by their factual popularity. When selecting sources to be included in a profile, we simulate this popularity by selecting popular URLs more often. These parameters enable the control of four important XML-elements contained in the semantic profiles that determine the system workload. These elements are: (i) *sources (URLs)*; (ii) crawling *depth* in each source; (iii) *keywords* and (iv) keyword *weights*. The output of this procedure is a set of semantic profiles which consist of a set of topics organized in a tree-like form. Each topic is composed of sources with corresponding keywords. Keywords are assigned an importance weight.

3.2 Execution Profiles Generation

An execution profile is partly derived from the corresponding semantic profile. Therefore, the semantic profiles created by the previous procedure are inputted to the *Execution Profile Generator*. We extract from semantic profiles, the information about preferred sources and form them into eRACE appropriate directives. These directives are encoded in XML in the form of Uniform Resource Description elements (URD). The URDs are produced by coalescing together all the sources defined in the semantic profile of a user. We also utilize a simple statistical model to generate directives on service-level QoS requirements that need to be specified in the execution profile. An example of such variable is the *update-frequency* attribute, which plays a vital role in the performance of the crawler. It represents how often a user wishes to receive an updated version of a page. If this value is small, the crawler is obliged to download the page often. Finally, we use default values for variables that do not play an important role in the crawler performance.

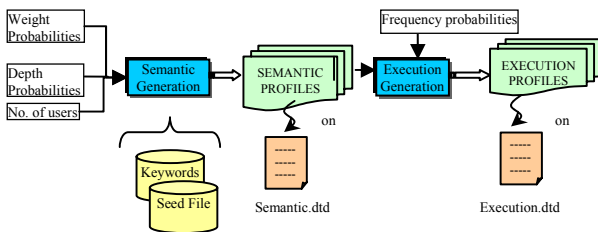


Figure 1: Toolkit overview

4. EXPERIMENTATION

We tested the profile generator against a varying number of users (starting from 1000 to 10000). The seed file we used consisted of 169237 URLs, the minimum and maximum number of URLs a profile can have is four and forty respectively. We measured the

time required to create the semantic and execution profiles and estimated the disk space needed. We observed that as the number of users grew, time and size required augmented linearly, which is a desired behavior. Indicatively, for 10000 users, we have about 19 sources in each profile and an average of 39 keywords in each profile; the majority of sources have a crawling depth of 0 and 1 which are the most popular values. It is self understood that we have repetition of some URLs since the total selected URLs are about 190000, while the total are only 169237.

With the purpose of ensuring that the tool creates a realistic environment, we created a chart of the chosen URLs against their popularity. We observed that this chart follows a Zipf distribution (Chart 1), which is expected according to Web characterization studies [3].

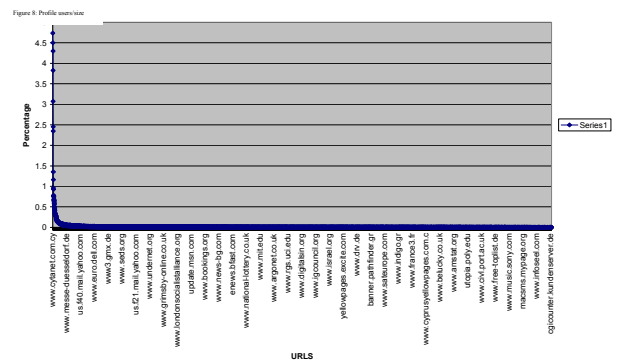


Chart 1: Zipf-like distribution of selected sources of user profiles

5. SUMMARY

Building a personalized Internet Service demands a careful design, a strong implementation and thorough experimentation. In this paper, we presented a profile generator tool that aims to simulate a realistic environment within which eRACE will be tested. The next step towards this direction is to create models that assess the results of the experimentation and identify bottlenecks that appear in the processes. The tool may also be used in the simulation of mobile users of the eRACE system.

6. ACKNOWLEDGEMENTS

This work was partly supported from the ANWIRE project and WebC-MINE project. Also special thanks to George Palioura and Hristo Floro from Demokritos Institute in Athens for their significant assistance.

7. REFERENCES

- [1] Open Directory Project: <http://dmoz.org/rdf>.
- [2] Google Web Service: <http://www.google.org/apis/>
- [3] "Web: Protocols and Practice," B. Krishnamurthy and J. Rexford, Addison Wesley.
- [4] eRACE Project webpage <http://www.cs.ucy.ac.cy/Projects/eRACE>