Proceedings of the 2011 Winter Simulation Conference
S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, eds.

MODELING WEB USAGE PROFILES OF CLOUD SERVICES FOR UTILITY COST ANALYSIS

Joseph Idziorek
Mark Tannian
Douglas Jacobson

Iowa State University
2215 Coover Hall
Ames, IA 50011, USA

ABSTRACT

Early proponents of public cloud computing have come to identify cost savings a key factor for adoption. However, the adoption and hosting of a web application in the cloud does not provide any such guarantees. This is in part due to the utility pricing model that dictates the cost of public cloud resources. In this work we seek to model and simulate data usage for a web application for the purpose of utility cost analysis. Although much research has been performed in the area of web usage mining, previously proposed models are unable to accurately model web usage profiles for a specific web application. In this paper, we present a simulation model and corresponding algorithm to model web usage based on empirical observations. The validation of the proposed model shows that the simulated output conforms to that of what was observed and is within acceptable tolerance limits.

1 INTRODUCTION

With the advent of the public cloud computing model, web services that were once hosted on private servers and networks are being outsourced to third-party cloud service providers (CSPs) - Amazon's EC2 is a well-known example. Early proponents of this emerging compute model have come to identify cost savings as a key motivation for the adoption of the cloud model. In comparison to more traditional computing models, economic efficiencies in the public cloud have been enabled by the fundamental paradigm shifts in the way computing infrastructure is hosted (e.g., multi-tenant hosting through virtualization, economies of scale, thin provisioning) and the pay-as-you-go business model that dictates costs for resource usages by the cloud consumer (i.e. one that rents computing infrastructure from a CSP) - namely, the utility compute costing model. However, the adoption and hosting of a web service in the cloud does not provide any guarantees of cost savings as there are many factors that must be taken into consideration.

Under the utility compute costing model, much like the utility model that governs the cost of electricity consumption, cloud consumers only pay for the resources they use and only for the time they use them. For instance, the web data transfer costs in and out of Amazon's EC2 environment by a cloud consumer's clients (those that patron the cloud-hosted web application) is governed by the Amazon's Web Services costing model (Figure 1) and accrues a cost that is a function of the total data transferred (Amazon Web Services 2011). At the conclusion of the month - a typical cloud billing cycle - the aggregated costs are billed to the cloud consumer. Because data usage in the cloud environment is uncertain, the cost for data transfer is as well, which is not typically the case for private web service hosting. The pay-as-you-go billing structure fundamentally changes how those who adopt the cloud model view the monthly data usage of their web applications and motivates the need for modeling and simulation of web usage profiles. Being able to accurately forecast accumulated resource consumption in advance allows one to anticipate costs and manage application designs in order to address costs proactively.

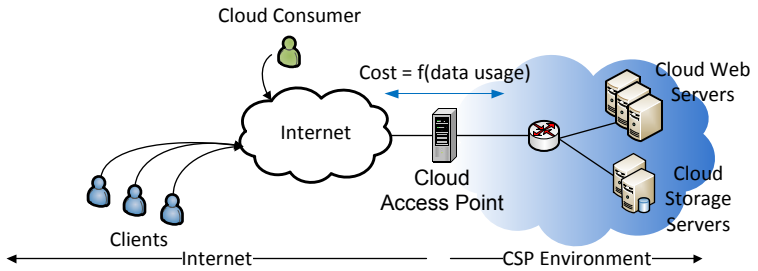

Figure 1: Cloud Network Diagram.

In this work we seek to model and simulate data usage for a web application for the purpose of utility cost analysis. More specifically, we seek to explore the minimum number of days of training data necessary to achieve acceptable accuracy of the simulation output. Although much research has been performed in the area of web usage mining - techniques to model and simulate web user transactions - previously proposed models that generate web traffic are unable to accurately model web usage profiles for a specific web application. Either these models generate generic web document requests or such requests are based on theoretical distributions. Our approach differs in that web document requests are derived from a Markov model trained on empirical observations of how the web application under consideration was used in actuality. In this paper, we present a simulation model and corresponding algorithm to model web usage based on empirical observations. The validation of the proposed model shows that the simulated output conforms to that of what was observed and is within acceptable tolerance limits.

The rest of the paper is organized as follows. Section 2 discusses related works in the context of this work. Section 3 describes the dataset used to train and validate our simulation algorithm as well as considerations taken when cleansing the original dataset. To model web usage profiles, in Section 4 describe our simulation modeling and corresponding algorithm. Based on this model, Section 5 provides the experimental metrics, design, and results used to validate the proposed model. Lastly, future work and a conclusion are discussed in Sections 6 and 7 respectively.

2 RELATED WORK

The related works that have bearing on this paper are derived from the research areas of web usage mining and web traffic generation. Although web usage modeling and traffic generation are not mutually exclusive, we will explore a shortcoming in the synthesis of these two research fields. At present, a complete model that takes into account all the necessary sub-models needed to accurately simulate realistic web traffic for a specific website does not appear to exist. Furthermore, there is no model suitable for predictive cost modeling of web traffic. In this section, we briefly describe our work in the context of these related bodies of work.

Much research has been performed in area of web usage mining since the seminal work done by Arlitt and Williamson (1996). Many of these works, similar to that of Yeung and Szeto (1999), Mah (1997), and Tran-Gia, Staehle, and Leibnitz (2001) have sought to characterize, model and validate the distributions that depict the way individual users and user populations interact with websites. Extrapolating from this body of research, a number papers have made use of and extended these models to simulate web traffic for a number of purposes. Cao, Cleveland, Gao, Jeffay, Smith, and Weigle (2004) presented a model to simulate generic web traffic on high-speed backbone links. Their objective differs from ours in that we seek to model aggregate user behavior for a distinct website as observed by the web server as opposed to modeling link traffic. Luo and Marin (2005) devised a model to simulate realistic Internet background traffic, including the web, for constructing a network intrusion detection environment. Similarly, Kroc, Eidenbenz, and Smith (2009) focused on modeling the theoretical distributions that together compose a

single web user session. While such modeling may be sufficient for background noise and generating realistic user-side web sessions respectively, neither of these two works model specific page requests as observed for a given website. Instead they model web interactions as generic requests. Moreover, the requested web document size is attributed a value based on a theoretical distribution, which is not sufficient for the purposes of accurately modeling actual data usage for a specific website with real document sizes. Instead, specific web requests need to be represented by their known data sizes, which is discussed in Section 3.1.

Burklen, Marron, Fritsch, and Rothermel (2005) present a general model and algorithm to synthetically generate a sequence of web requests for a single user. This work is based on known and previously studied web usage behaviors and models in addition to the hyperlink structure of individual web pages and their relationship to other web pages for a given website. While notionally similar to our work, the scope of such an algorithm is limited to that of a single user session, not multiple users over a prolonged period of time, which is a key objective of this work. Furthermore, the synthesizing of a single user session is predicated on a theoretical relationship between web pages derived from the site's hyperlink structure and not from leveraging historical observations of how users have traversed the website. We instead generate individual requests that compose a web session from a Markov model based on learned browsing patterns.

In contrast to modeling generic or request sequences based hyperlink structures, Markov chains have been shown to provide accurate models for simulating web usage (Li and Tian 2003). Under this guise, Markov models have been used in a number of contexts including performance analysis (Cheng, Chang, and Zhang 2007) and caching algorithms (Chen and Zhang 2003). Most similar to our work are papers that have sought to predict user web sessions by means of Markov models. Nigam and Jain (2010) presented a model based on a dynamic nested Markov model for predicting the next page accessed by a user given an observed series of requests. Borges and Levene have produced a number of works - summarized in Borges and Levene (2007) - that investigate the next page request of individual users and the accuracy of predicting n-grams of requests with various Markov models. While effective for their given purposes, neither of these works provide analysis of the accuracy and summarization ability of using Markov models for generating and predicting aggregate web traffic based on actual server logs.

In their own respective way, each of these works presented in this section falls short of being able to provide a complete model and simulation framework for data usage transferred by a specific website. In this paper, we have created a synthesis between many of the sub-models presented in these works to provide a more relevant modeling of content usage for a specific website based on training from session logs. This model can then be used for the purposes of modeling usage profiles of cloud-based service for predictive cost analysis.

3 DATASET DESCRIPTION AND CONSIDERATIONS

The dataset is a 55-day web server log originating from our department's web server and is used to both train the proposed model as well as for the experimental validation of the simulation algorithm presented in Section 4.

Web server usage is often represented as zipf-like distributions (Breslau, Cao, Fan, Phillips, and Shenker 1999) in which the frequency of a requested document $p(i)$ is proportional to its rank i such that $p(i) \propto 1/i^{\alpha}$ where α is close to unity. Figure 2 depicts a zipf-like distribution for the given dataset as a log-log plot of request frequency vs. rank. Figure 3 shows a similar log-log plot of document rank vs. data usage. Drawing from Figure 2, the 356 most requested pages represent 90% of all requests and of these pages their weight in data usage totals over 97% of data requested (Figure 3) over the observed 55-day span. Given these empirical distributions and observations, the modeling of data usage for the given dataset is heavily dependent upon accurately modeling the most frequently requested documents.

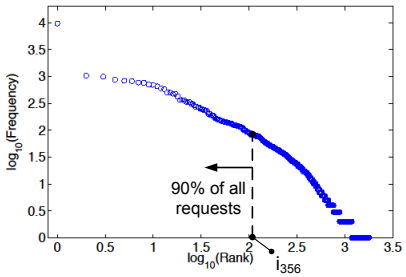

Figure 2: Request Frequency.

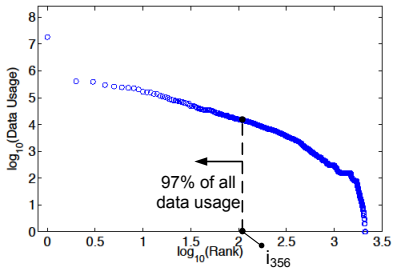

Figure 3: Data Usage.

3.1 Web Usage Mining and Modeling Components

A web server log maintains an itemized journal of all users' content requests and provides the necessary observations for deriving empirical distributions and models used in the study of web usage. Figure 4 provides an illustration of web usage metrics and will be used as a guide to explain these metrics as well as the considerations taken to cleanse the observed web server log from its original form to what was used for the purposes of this paper.

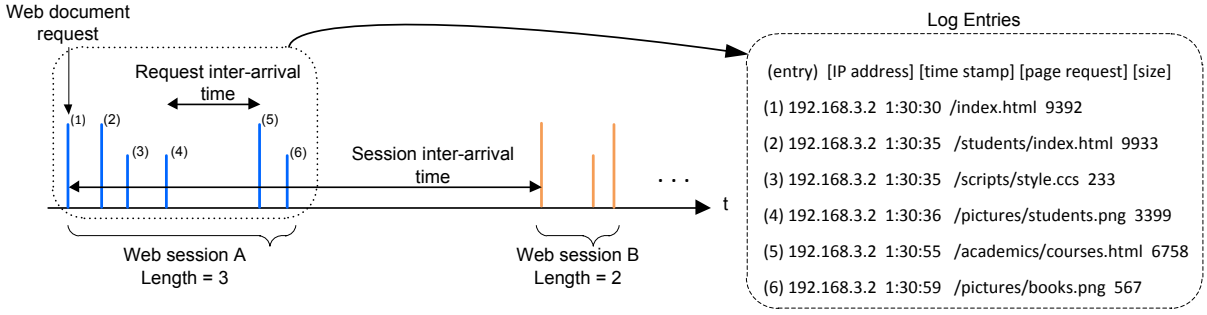

Figure 4: Web Usage Modeling Components.

Primary Web Document Request - An individual web request within a web log is depicted as a vertical line in Figure 4. As seen annotated in the call-out, requests are composed of an IP address for the requesting client, time stamp, document requested, and the data size of the respective document.

The modeling and simulation objectives of this work are reliant on the accuracy of the data size attributed to each client-invoked request (i.e. a primary request). The data size, as shown in Figure 4, for each individual HTML request can be misleading as it is not a complete account of the data usage needed to view the expected web page, but instead the entry only reports the size of the HTML file itself. Typically, a single request for a HTML document invokes other secondary in-line requests to retrieve embedded objects within the primary HTML page such as pictures, scripts, and videos. In Figure 4, both $(3)/scripts/style.css$ and $(4)/pictures/students.png$ are secondary in-line requests, shown as shorter lines, of the primary request, shown as a taller line, for $(2)/students/index.html$. Together, the size of the primary and associated secondary requests represent the total data usage for a single client-invoked request, which is the objective of this analysis. While both primary and secondary requests are registered in a web log, archival analysis is inadequate to determine whether an entry is a primary or a secondary request, and relate secondary requests with its parent primary request. Client-side and distributed caching further complicate the task of reconstructing these relationships from a web log.

Therefore, in lieu of these impediments and in order to accurately capture the data size for each primary client-invoked request and its accompanying secondary requests, analysis was performed on the active departmental website. The URL for each primary request in the web log - assumed to be an HTML

document - was requested from the website with a script capable of capturing the data usage footprint for the primary request as well as all secondary requests. After accumulating the footprint of each primary request and relevant secondary requests, the total replaced the original primary request size and the secondary entries were discarded. Such analysis and post-processing can only be performed with access to a live website. The presented initial results are limited to that of a single website, since access to web logs of an active site is severely limited.

Web Session - A web session is a set of consecutive requests generated by an individual user during a single viewing period. As seen in Figure 4, *web session A* contains three primary web requests and thus has a web session length of three. A web log is composed of many interleaved web sessions initiated by multiple users. Within the observed web log described in this section, web session lengths ranged from well over 100 documents in length with some sessions as long as 1400 primary requests. In order to provide a more accurate modeling of how the majority of normal users actually traversed the website, web session lengths were truncated to 35 primary requests, which falls within the 99th percentile.

Often web logs do not contain the complete information necessary to discern when a web session for a user ends and when the user's next web session begins. Research in this area has sought to differentiate between sessions using time-oriented heuristics (Zhang and Ghorbani 2004) and transitional request probabilities using a first-order Markov model (Požandenel, Mahnič and, and Kukar 2010). However, for simplicity, it is assumed in this work that a 900 second or greater time lapse between primary requests denotes the end of one web session and the beginning of a new session. This assumption is consistent with previous works in the field (Kroc, Eidenbenz, and Smith 2009).

Session Inter-arrival Time - Session inter-arrival time is the measure of time between the beginning of two consecutive web sessions. In Figure 4, the session inter-arrival time is depicted as the time between the beginning of *web session A* and that of *web session B*.

Request Inter-arrival Time - Within a given session greater in length than one, there is an intermittent amount of time experienced between each respective primary web request by the client. This is referred to as the request inter-arrival time and is shown between *web session A* requests *(4)* and *(5)* in Figure 4.

3.2 Dataset Limitations

As with many empirically based models, the simulation results are heavily dependent on the quality of the training data. Moreover, the training of Markov models based on the actions that have been observed in the web logs restricts our model to only the web pages requested and conditional probabilities between pages that have been observed. Lastly, due to the necessity of performing analysis on a live website and limited cooperation of website operators, the results of our work are currently limited. Although there are no indications that the presented model and algorithm would not provide a general solution, such claims can only be made after further analysis of a broader set of websites.

4 SIMULATION ALGORITHM

The objective of the proposed simulation algorithm is to generate web traffic in accordance with what has been observed. The uniqueness of the outlined approach in comparison to the papers discussed in Section 2 is that the described simulation algorithm generates web traffic crafted from empirical distributions from a specific web application and utilizes a trained Markov model to generate page requests that reflect actual primary request patterns. This section describes the proposed algorithm - based on a second-order Markov model - and the underlying modeling components.

4.1 Algorithm Description

Given a web server log composed of N days of observed requests as an input, the objective of Algorithm 1 is to simulate a web server log that conforms to the empirical distributions derived from the input dataset while preserving web usage behaviors as they were deduced from sessions within the input web log. The

output of the simulation algorithm is a web server log L composed of many users' web sessions that emulates actual clients as they utilize the website over a period of time.

A web server log L is composed of many independent and, at times, overlapping sessions s that represent the actions taken by a website's user base. Each session $s \in L$ is a tuple $s =< ipAddress, sessionLength, P >$ that is composed of the IP address of the individual requester, the number of web pages requested during a given session, and the set of primary web page requests P. Each $p \in P$ is also a tuple $p =< page, time, size >$ that denotes the specific web page, time stamp, and size of each individual web page request within a session.

Algorithm 1 Modeling Web Usage from a Second-Order Markov Model

Input: Observed Web Server Log
Output: Generated Web Server Log

```
1:  generateLog() : L {
2:  absoluteTime ← 0, i ← 0, currentRequests ← 0;
3:  while currentRequests < totalRequests do
4:     s_i.sessionLength ← generateSessionLength();
5:     s_i.ipAddress ← generateIpAddress();
6:     absolute_time += generateSessionInterarrivalTime();
7:     currentRequests+ = s_i.sessionLength;
8:     for j ← 1 to s_i.sessionLength do
9:        if j == 1 then
10:          p_j.page ← returnFirstPage();
11:          p_j.time ← absoluteTime;
12:          relative_time ← absoluteTime;
13:       else
14:          if j == 2 then
15:             p_j.page ← returnPageFirstOrderMarkov(p_{j−1}.page);
16:          else
17:             p_j.page ← returnPageSecondOrderMarkov(p_{j−1}.page, p_{j−2}.page);
18:          end if
19:          relativeTime += generateRequestInterarrivalTime();
20:          p_j.time ← relativeTime;
21:       end if
22:       p_j.size ← pageSize();
23:    end for
24:    L = L ∪ s_i;
25:    i++;
26: end while
27: return L;
28: }
```

4.2 Algorithm Modeling Components

The presented algorithm illustrates a high-level overview of our approach to simulating web usage profiles. The underpinnings of the algorithm are derived from published web usage metrics in coordination with generating primary request sequences by using a Markov model. The following descriptions provide a

thorough analysis of the algorithm components used for experimental evaluation, which is reviewed in the next section.

Line 3: *totalRequests* - Although not formally presented in in this paper, linear regression analysis was performed on each training dataset to extrapolate the expectation of the number requests for the given target of accumulated simulation days. This value was used as the control parameter to dictate the length of each simulation run.

Line 4: *generateSessionLength*() - The session length defines the number of primary web requests by an individual user during a single browsing period. Session lengths were modeled as a Lognormal distribution with the following parameters: α = 0.44-0.48, β = 0.76-0.78, μ = 2.47-2.49 pages, σ = 3.45-3.46 pages. The modeled session length distribution is consistent with Tran-Gia, Staehle, and Leibnitz (2001).

Line 5: *generateIpAddress*() - The IP addresses chosen for each individual session were modeled and drawn from a continuous, piecewise-linear empirical distribution that was populated based on the pre-processing analysis of the training data set.

Line 6: *generateSessionInterarrivalTime*() - The session inter-arrival times were modeled as an exponential distribution with a mean that varied between 113 and 126 seconds. Although in Kroc, Eidenbenz, and Smith (2009), session length was modeled with a Weibull distribution, we found an exponential distribution to be a more appropriate fit for the given data sets.

Line 10: *returnFirstPage*() - Each simulated session is initialized by determining the first page to be synthetically generated for a given user and for each individual session. The first page distribution is an initial state vector learned from the first page views of the sessions extracted from the training dataset. Due to the self-similarity of web traffic, the distribution of first page requests can also be depicted with a zipf-like distribution similar to that of the aggregate request distributions presented in Section 3. Figure 5 represents the first-page distribution of the training data set and Figure 6 represents the simulated first-page distribution.

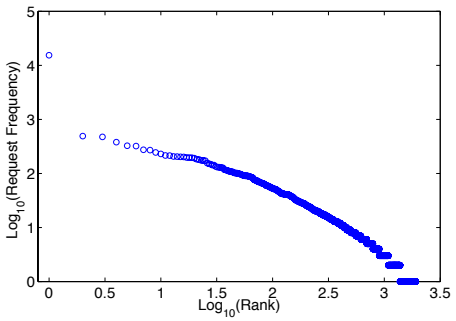

Figure 5: Actual First-Page Zipf Distribution.

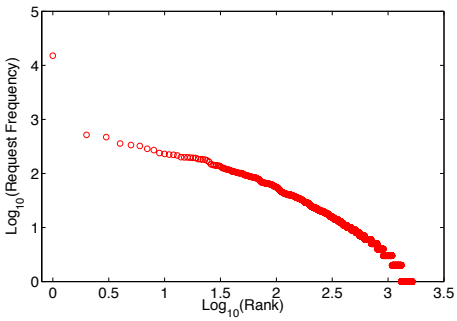

Figure 6: Generated First-Page Zipf Distribution.

Line 15: *returnPageFirstOrderMarkov*() - A Markov model is used to generate the actual page requests and is trained by analyzing web server logs. Based on the presented algorithm, if a session length is at least two, the second request generated in a web session is drawn from a first-order Markov model such that $p_{ij} = Pr(x_{n+1} = j | x_n = i)$. For a thorough explanation of training and building Markov models, which is accompanied examples, please see Borges and Levene (2007).

Line 17: *returnPageSecondOrderMarkov*() - For web session lengths greater than two, all subsequent requests are generated from a second-order Markov model such that $p_{ijk} = Pr(x_{n+2} = k | x_{n+1} = j, x_n = i)$. The second-order model was constructed in a manner similar to the first-order model.

Line 19: *generateRequestInterarrivalTime*() - Each request inter-arrival time was generated from a Weibull distribution with the scale parameter ranging between 28.57 and 33.8 and the shape parameter

between 0.57 and 0.62. The fitting of this distribution aligns with that in Kroc, Eidenbenz, and Smith (2009).

5 EXPERIMENTAL EVALUATION

In this section we present experimental measures, experimental design and related experimental results. Validation of the algorithm relies on three key measures : 1) summarization quality, 2) overlap and 3) data usage modeling error that are assessed relative to actual web usage recorded by the website.

5.1 Experimental Metrics

The Spearman's Footrule distance (Dwork, Kumar, Naor, and Sivakumar 2001) is a non-parametric measure of association between two ranked lists. This measure was used in a similar context to measure the accuracy of predicting individual session n-grams generated from Markov models (Borges and Levene 2007). For our work, however, the Spearman's Footrule distance is instead utilized to measure the summarization accuracy of the simulation algorithm by analyzing the ranked lists of the top-10% of the simulated primary requests output in comparison to that of the web logs.

The purpose of employing the Spearman's Footrule is to find an aggregated ranking that minimizes the distance between two ranked lists. However, for the purposes of this paper, the proximity between two ranked lists will instead be considered as it is a more appropriate measure that aligns with the described analysis objectives.

Drawing on the notation established in Borges and Levene (2007), the Spearman's Footrule proximity is defined as follows: Given two ranked top-k lists L_1 and L_2 as inputs, with each list containing k entries, let L be the union of the two lists such that $L = L_1 \cup L_2$. Furthermore, let L_1 be the reference list that is assumed to be the ground truth and L_2 be the comparison list, which in all actuality is a partial list in comparison to that of the reference list. To obtain the ranking of a list item $i \in L$ in L_1, we define the function $f(i)$ and similarly $g(i)$ for $i \in L_2$. In either function $f(i)$ or $g(i)$ if $i \notin L$, then the subsequent ranking is assigned that of a location parameter $l = k+1$ (Sculley 2007). Given these preliminaries, the Spearman's Footrule proximity is defined as follows:

$$F(L_1, L_2) = 1 - \frac{\sum_{i \in L} |f(i) - g(i)|}{k(k+1)} \tag{1}$$

In the case that both ranked lists were identical, the Spearman proximity would be one. In order to provide a measure of similarity or proximity instead of a measure of difference or distance, the normalized summation in Equation 1 is subtracted from 1.

In conjunction with the Spearman's Footrule proximity, the overlap between the reference list L_1 and the comparator list L_2 is measured to provide a broad indication of the summarization ability of the simulation model output. The overlap is defined as the percentage of items in the comparator list L_2 that appear in the reference list L_1.

Lastly, the percent error between the expected value of the aggregate data usage (in bytes) as produced by the output from the simulation model and the actual data usage from that of the observed logs is measured. This measure compliments the Spearman's Footrule and overlap metrics to provide a comparative indication of the model's ability to accurately forecast aggregate data usage, which is a key variable in public cloud utility costing models.

5.2 Experimental Design

The available dataset was utilized to its full extent by using a sliding window for both the input training days and for the output comparison. A goal of this work was to explore the minimum number of training days necessary to accurately simulate future aggregate use. In the context of cloud computing and due to

monthly billing cycles, future aggregate use is most appropriately defined as the 30 days in advance of the first observed day of the given training window. For example, as illustrated in Figure 7, the *simulation run A* is trained on the days 1-10 of accumulated logs and is tasked with simulating the aggregate usage of the web application from days 1-30. The results of such a simulation output are then compared to that of the observed web log. To provide multiple simulation runs for a given training window size, the simulation was repeated by shifting the simulation window into the future one day. As shown in Figure 7, the *simulation run B* was trained on the logs from days 2-11 and was tasked with simulating the aggregate usage for days 2-31 and thus a sliding training window of 10 days.

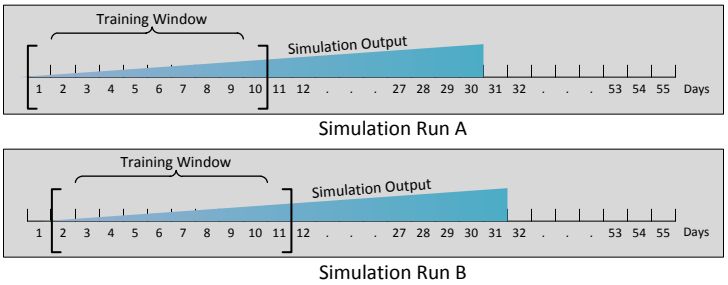

Figure 7: Experiment Simulation Design.

To explore the minimum number of training days necessary to simulate future aggregate use, a sliding window of observed days ranging from 4 days to 29 days was used. The size of the training window (in days) is denoted as the x-axis for Figures 8, 9, and 10. For each training window size, 25 simulation runs were conducted and the metrics described in Section 5.1 were calculated. Each datapoint on Figures 8, 9, and 10 represents an average for the given metric resulting from 25 simulations runs per training window size.

5.3 Experimental Results

Experimental simulations were performed for the described algorithm using both a first-order and second-order Markov model to generate the actual web page requests that compose a web session. Having prior access to the accumulated data logs allowed for the comparison of the simulation model output with that of what actually transpired.

Figure 8 provides a comparison of percent error between generating web requests from a first-order Markov model and from that of a second-order Markov model. From this figure it can be seen that at least nine days of observed web logs are necessary to provide a 30-day projection of data usage that is within a 5% error tolerance of the actual value. As expected, the initial prediction capability of both models improves as the training window sizes increases. However, the first-order model reaches a limitation of accuracy after approximately 11 days while the second-order model exhibits a linear improvement in accuracy after 10 days with a clear divergence between the two models occurring after 18 days.

Figure 9 shows the Spearman's Footrule proximity between the simulated output and the observed logs for the top-10% of requests. The results show that across all training window sizes, the first-order model provides a consistently better summarization ability in comparison to that of the second-order model. This is in part due to the nature of Markov models. In this context, first-order models accurately represent the first two requests of a session but do not accurately represent all second-order conditional probabilities for session lengths greater than two. Second-order models on the other hand, accurately model second-order conditional probabilities and thus have a higher accuracy in reflecting reality (Figure 8) but do so at the loss of coverage. The summarization strengths of the first-order model in comparison to the second-order model can further be seen in Figure 10, which provides a higher-level comparison of the overlap between the top-10% of requests.

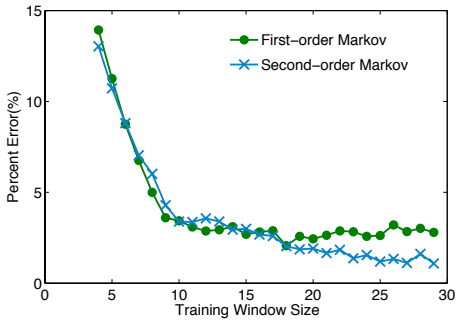

Figure 8: Data Usage Percent Error.

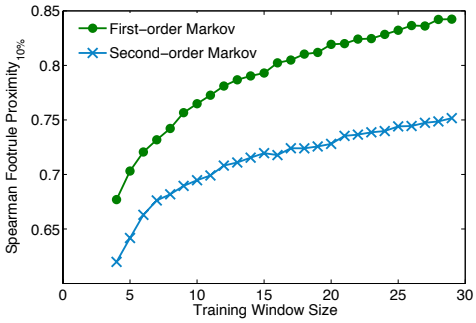

Figure 9: Spearman's Footrule Proximity.

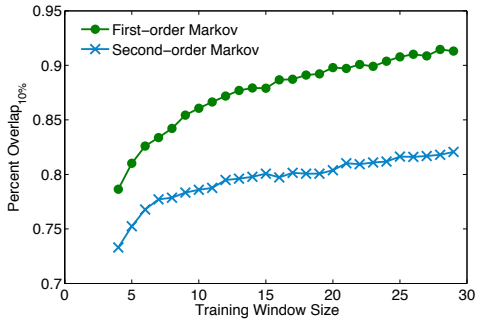

Figure 10: Overlap Percentage.

Through this study, a second-order Markov model has been shown to provide a consistently more accurate modeling of web data usage than a first-order model. However, a first-order Markov model has been shown to consistently better coverage in comparison to a second-order model. The drawback of higher-order Markov models is that they consume a exponentially higher state-space and longer model runtimes than that of lower-order models. For the application of data usage forecasting a second-order model appears to be preferable. However, for other applications of our model where coverage is a more valued quality, limiting modeling to the use of first-order Markov model may be preferable.

6 FUTURE WORK

Further validation and future work on this topic will be heavily dependent on obtaining a sufficient number of daily web logs from an active websites of sufficient length and that are suited for data usage modeling. Privacy and security concerns are significant impediments to accessing web logs for research use. To render existing publicly available datasets useful for such analysis, heuristic-based approaches will be necessary in order to identify primary requests and to reconstruct data usage footprints from associated secondary in-line requests. Within the study of Markov models, there exist a number of methodologies that could be implemented (or expanded upon) to further refine the accuracy of session generation component of the model, which could potentially lead to more precise data usage modeling.

7 CONCLUSION

Resource planning is not unique to where an application is hosted. However, having accurate profiles of future web application usage allows for more efficient management and expectations of costs in the cloud. In order to address this challenge, modeling needs to be able to characterize a given web application trained with actuals usage patterns. In summary, our approach was to do the following: 1) obtain actual page

sizes (i.e. size of primary and secondary in-line requests); 2) determine empirical distributions from actual web logs; 3) build Markov models that represent page request order based on actual usage; 4) apply an algorithm that leverages the models in 2) and 3) to generate web log entries for the target number of days; and 5) evaluate accuracy and summarization of the resultant logs compared with actual logs. In a practical setting the last step would not be possible until after the target day has passed, but step 5) would be helpful to establish ongoing confidence in and possibly tune parameters within the approach.

This paper contributes to the field of modeling and simulation by offering a modeling approach that approximates actual web application usage behavior with greater fidelity by tailoring modeling to the application instance. Moreover, the developed algorithm provides a means to utilize these models to produce days of complete web server logs. Thus providing one an ability to evaluate the qualities of this approach with a practical benchmark. Results have shown that a minimum of nine days of observed logs is necessary to provide a sufficiently accurate projection of logs for a cloud billing cycle. Ultimately a simulation algorithm that utilizes higher order Markov models yields acceptable data usage error rates at the expense of coverage, state space size and execution duration.

ACKNOWLEDGMENTS

The authors would like to thank the proceeding editors and the anonymous reviewers of this work.

REFERENCES

Amazon Web Services 2011, November. "Amazon EC2 Pricing". http://aws.amazon.com/ec2/pricing/.

Arlitt, M. F., and C. L. Williamson. 1996, May. "Web server workload characterization: the search for invariants". *SIGMETRICS Perform. Eval. Rev.* 24:126–137.

Borges, J., and M. Levene. 2007, April. "Evaluating Variable-Length Markov Chain Models for Analysis of User Web Navigation Sessions". *IEEE Transactions on Knowledge and Data Engineering,* 19 (4): 441–452.

Breslau, L., P. Cao, L. Fan, G. Phillips, and S. Shenker. 1999, March. "Web caching and Zipf-like distributions: evidence and implications". In *Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE INFOCOM '99.*, edited by IEEE Communications Society, Volume 1, 126–134.

Burklen, S., P. Marron, S. Fritsch, and K. Rothermel. 2005, April. "User centric walk: an integrated approach for modeling the browsing behavior of users on the Web". In *Proceedings. 38th Annual Simulation Symposium, 2005.*, edited by IEEE Computer Society, 149–159.

Cao, J., W. Cleveland, Y. Gao, K. Jeffay, F. Smith, and M. Weigle. 2004, March. "Stochastic models for generating synthetic HTTP source traffic". In *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM 2004.*, edited by IEEE Communications Society, Volume 3, 1546–1557.

Chen, X., and X. Zhang. 2003, Mar. "A popularity-based prediction model for Web prefetching". *Computer* 36 (3): 63–70.

Cheng, S., C. Chang, and L.-J. Zhang. 2007, July. "Stochastic Modeling Study for Competitive Web Services Market". In *IEEE International Conference on Web Services, 2007. ICWS 2007.*, 960–967.

Dwork, C., R. Kumar, M. Naor, and D. Sivakumar. 2001. "Rank aggregation methods for the Web". In *Proceedings of the 10th international conference on World Wide Web*, WWW '01, 613–622: ACM.

Kroc, L., S. Eidenbenz, and J. Smith. 2009, December. "SessionSim: Activity-based session generation for network simulation". In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 3169–3180. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Li, Z., and J. Tian. 2003, Nov. "Testing the suitability of Markov chains as Web usage models". In *27th Annual International Computer Software and Applications Conference, COMPSAC 2003.*, 356–361: IEEE Computer Society.

Luo, S., and G. Marin. 2005, December. "Realistic Internet traffic simulation through mixture modeling and a case study". In *Proceedings of the 2005 Winter Simulation Conference*, edited by M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 9 pp. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Mah, B. 1997, Apr. "An empirical model of HTTP network traffic". In *Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM '97.*, Volume 2, 592–600.

Nigam, B., and S. Jain. 2010, Nov.. "Generating a New Model for Predicting the Next Accessed Web Page in Web Usage Mining". In *2010 3rd International Conference on Emerging Trends in Engineering and Technology (ICETET)*, 485–490.

Požandenel, M., V. Mahnič and, and M. Kukar. 2010, Dec. "Separation of Interleaved Web Sessions with Heuristic Search". In *2010 IEEE 10th International Conference on Data Mining (ICDM)*, 411–420.

Sculley, D. 2007. "Rank Aggregation for Similar Items". In *Proceedings of the Seventh SIAM International Conference on Data Mining*: SIAM.

Tran-Gia, P., D. Staehle, and K. Leibnitz. 2001. "Source Traffic Modeling of Wireless Applications". *AEU - International Journal of Electronics and Communications* 55 (1): 27–36.

Yeung, K., and C. Szeto. 1999. "On the modeling of WWW request arrivals". In *1999 International Workshops on Parallel Processing*, 248–253.

Zhang, J., and A. Ghorbani. 2004, May. "The reconstruction of user sessions from a server log using improved time-oriented heuristics". In *Second Annual Conference on Communication Networks and Services Research, 2004*, edited by A. A. Ghorbani, 315–322.

AUTHOR BIOGRAPHIES

JOSEPH IDZIOREK is a PhD Candidate in the Department of Electrical and Computer Engineering at Iowa State University in Ames, IA, USA. His research interests broadly lie in the areas of cloud computing security, anomaly detection, distributed denial of service attacks and modeling and simulation. He obtained his BSc in Computer Engineering from St. Cloud State University in St. Cloud, Minnesota, USA. His email address is idziorek@iastate.edu.

MARK TANNIAN is currently a PhD candidate at Iowa State University, Iowa, USA in Computer Engineering with interests in cloud computing security and information security visualisation. He returned to pursue his PhD after 12 years of professional experience in information security and holds the CISSP credential. His professional experiences range from technical firewall support, consulting security engineer, technical product manager, senior operations security analyst, product trainer and technical sales engineer. He obtained his Bachelor of Electrical Engineering from University of Delaware, Delaware, USA and his Masters of Electrical Engineering from George Washington University, Washington DC, USA. His email address is mtannian@iastate.edu.

DOUGLAS JACOBSON is a University Professor in the Department of Electrical and Computer Engineering at Iowa State University. Dr. Jacobson joined the faculty in 1985 after receiving a Ph.D. degree in Computer Engineering from Iowa State University in 1985. Dr. Jacobson is currently the director the Iowa State University Information Assurance Center. Dr. Jacobson teaches network security and information warfare and has written a textbook on network security. Dr. Jacobson has received two R&D 100 awards for his security technology and has two patents in the area of computer security. Dr. Jacobson has given over 50 presentations in the area of computer security and has testified in front of the U.S. Senate committee of the Judiciary on security issues associated with peer-to-peer networking. His email address is dougj@iastate.edu.