# Modeling Object Characteristics of Dynamic Web Content

Weisong Shi, Eli Collins, and Vijay Karamcheti
Department of Computer Science
Courant Institute of Mathematical Sciences
New York University
{*weisong, vijayk*}*@cs.nyu.edu*

## Abstract

Although requests for dynamic and personalized content increasingly dominate Internet traffic, there is an absence of (1) good models describing characteristics of dynamic web content, and (2) synthetic content generators, which can be used for simulation-based study of new techniques.

This paper addresses both of these shortcomings. Its primary contribution is a set of models that capture the characteristics of dynamic content at the sub-document level in terms of independent parameters such as the distributions of object sizes and their freshness times, and derived parameters such as content reusability across time and linked documents. A secondary contribution is a publicly available Java-based dynamic content emulator, DYCE, which uses these models to generate edge side include (ESI) based dynamic content in response to requests for both whole documents and separate objects.

## 1 Introduction

To efficiently serve and deliver dynamic and personalized content, researchers have proposed several server-side and cache-side mechanisms. Server-side techniques such as delta encoding [18], data update propagation [9], and fragment-based page generation [10, 14], reduce the burden on the server by allowing reuse of previously generated content to serve new requests. Cache-side techniques, exemplified by systems such as Active Cache [8], Gemini [19], CONCA [23], and Wills et al.'s [26] content assembly technique, attempt to reduce the latency of dynamic content delivery by moving some functionality to the network edge. Similar trends are also visible in commercial products such as IBM's WebSphere [15] and Akamai's Edgesuite [1]. These approaches all view the document in terms of a quasi-static *template* (expressed using formatting languages such as XSL-FO [28] or edge-side include (ESI) [24]), which is filled out with multiple individually cacheable and/or uncacheable *objects*. This object composition assumption enables surrogates and downstream proxy caches to reuse templates and cached objects to efficiently serve subsequent requests and additionally reduces server load, bandwidth requirements and user-perceived latencies by allowing only the unavailable objects to be fetched.

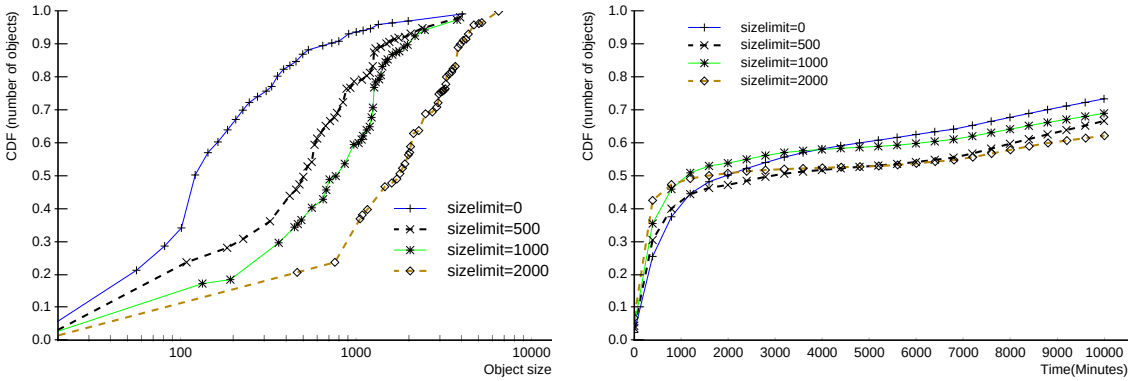
Although the above techniques appear promising, it is difficult to predict to what extent their stated benefits apply to a workload different than the one they were evaluated on. Consider the following questions that we were faced with when designing general policies for our CONCA architecture [23]: At what granularity must objects be cached to achieve sufficient reuse? Is their a sharp threshold for choosing this granularity, or is it the case that the benefits are continuously varying? Can we estimate likely freshness times of objects from the duration they have been present in the cache? Is their a correlation between object size and their reuse? While trying to answer these questions we encountered the problem that there is an absence of models for dynamic content. Even if such models were present, an additional problem one encounters while trying to evaluate a new technique is the absence of template-based content.

This paper addresses both shortcomings. By analyzing the content of six web sites that serve dynamic content over a two week period, we derive a set of models that characterizes this content in terms of a small number of independent parameters. Our studies find that the sizes and freshness times of component objects can be captured very well using *Exponential* and *Weibull* distributions respectively, and demonstrate siginificant content reusability across both the temporal and spatial dimensions. These models enable us to design and implement an effective Java-based dynamic content emulator (DYCE), which generates parameterizable dynamic content that adheres to the ESI specification.

The rest of this paper is organized as follows. Section 2 describes the methodology used in this study, specifically the extraction of information about document templates and component objects. Using one of the sites as a representative example, the analysis results are presented in Section 3. Section 4 describes the design, and implementation of the stand-alone dynamic content emulator. Section 5 discusses related work and we conclude in Section 6.

## 2 Methodology

Surrogates and proxy caches that are attempting to improve delivery of dynamic content, are interested both in independent parameters, such as *size distribution* of the objects that make up a document, and their *freshness times* (the length of time the object remains valid), as well as derived param-




(a) size distribution (b) freshness time distribution

Figure 1: The measured cumulative distribution of object sizes and freshness time in the `cnn` trace for different settings of size limit: (a) size, (b) freshness time.

eters, such as *content reusability* across repeated requests for the same document (temporal), and across requests for other documents that are linked from this one (spatial). Content reusability is defined to be the fraction of the document that can be reused, and quantifies the potential benefits from caching or reuse at the sub-document granularity.

Obtaining the above metrics is straightforward if one had access to the content provider web server and/or application server. Unfortunately, most commercial web sites which create dynamic content are proprietary. Therefore, we adopt an alternate approach based on the passive analysis of HTML content downloaded from various web sites. Our approach includes four steps: (1) *Data Collection*; (2) *Tree Building* to extract the template associated with the document, and identify the (logical) objects that fill out this template by grouping neighboring objects that exhibit similar characteristics; (3) *Tree Comparison*, which compares two documents across time, identifying the *freshness time* of each object; and (4) *Object Grouping*, which aggregates logical objects into physical objects that serve as the granularity at which document characteristics are modeled.

Two difficulties must be overcome in this approach. First is the lack of an explicit template associated with the document, which can help identify its component objects. To work around this difficulty, we make the assumption that *the document template can be expressed as a nested table*, an assumption that is true of most sites.

The second difficulty is that the object grouping step should ideally take into consideration document semantics such as the relationship of objects with each other. In the absence of such information, we have to use heuristic means. In this paper, we use two techniques to split the object tree generated into component objects: *size-based* and *level-based*. Size-based splitting chooses nodes whose size exceeds a certain threshold (and whose child nodes have size smaller than the threshold). Level-based splitting follows the logical structure of the document: all nodes below a certain depth in the tree are grouped together. More details of the methodology can be found in a technical report [22].

Using this methodology, we analyzed traces collected from three news sites (`www.cnn.com`, `dailynews.yahoo.com`, `www.nytimes.com`), two e-commerce sites (`www.amazon.com`, `www.barnesnoble.com`), and an entertainment site (`www.windowsmedia.com`). The main pages at these sites were downloaded every ten minutes over a two-week interval from January 31 to February 14, 2002.

# 3 Dynamic Content Characteristics

Due to space restrictions, we discuss only the results for the `cnn` trace, which is representative of the others.

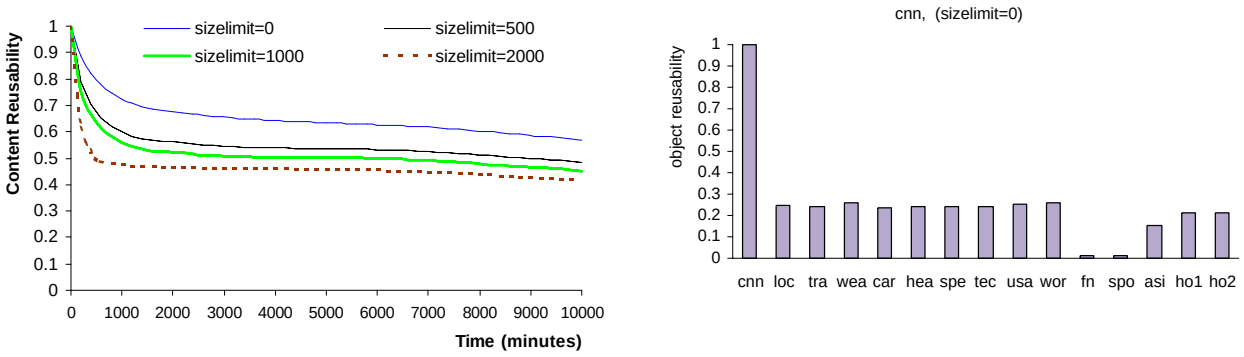
## 3.1 Object Sizes and Freshness Times

Figure 1 shows the measured cumulative distribution of object sizes and freshness times for different size-limit settings; the latter parameter denotes the target of the document splitting process outlined earlier. We find that a large number (about 90%) of objects in our documents are of relatively small size (smaller than 500 bytes), and while half of the objects exhibit freshness times that fall between a few minutes and a day, a significant fraction (almost 50%) of the objects are relatively long-lived, remaining essentially unchanged over the duration of the trace.

To develop models for the object size and freshness time distributions seen for different size limits, we use standard statistical methods similar to those used by Paxson in [21].

Comparing several distributions, such as Lognormal, Exponential, Weibull and Pareto, using the Chi-Square method as the goodness-of-fit [12] measure, we found that object sizes were best modeled using an *Exponential* distribution (with CDF $F(x) = 1-e^{-\lambda x}$). The results of the Chi-Square tests were in the range 0.1 to 2.5, showing a very close fit. Although not shown, we observed similar distribution fits for different settings of the level limit.

For the news and media sites, we found that except for a considerable fraction of objects that change very infrequently (see Figure 1(b)), the freshness times of component objects can be captured using a *Weibull* distribution (with




(a) time (b) spatial

Figure 2: Content reusability of `cnn` trace: (a) across time dimension, and (b) across linked documents.

CDF $F(x) = 1 - e^{-(\lambda x)^b}$), a variant of the Exponential distribution. For the two e-commerce sites, this distribution degenerates into a sharp bimodal one: *all of the objects either change on almost every access or change very infrequently.*[1]

To ascertain if there was a relationship between the sizes and freshness times of objects in the documents, we computed the correlation coefficient for the two series. We found that for our traces, there was no conclusive indication of any correlation between the two metrics.

## 3.2 Content Reusability

Figure 2(a) shows the content reusability (in terms of the bytes in the document that can be reused) over a one-week interval for the homepages of our six web sites. A few points need to be made here. First, each curve is an average of 1008 contiguous documents (corresponding to a one-week trace spaced at 10 minute intervals), each of which has been compared with 1008 subsequent documents in the series. Second, the level limit was set to the highest possible in each case to ensure that we obtained the maximum potential content reusability. Finally, the curves include reuse of the template, which occupies between 10%–20% of the whole document. As can be seen, there is considerable reuse.

Figure 2(b) shows the content reusability between the default `cnn` homepage and fifteen other linked documents: `local`, `travel`, `weather`, `career`, `health`, `special`, `technology`, `US`, `world`, and the top two headlines, all of which are hosted on the CNN server, as well as two links `sports` and `asia`, hosted on different servers. The figure shows that 25% of the document bytes can be shared across other documents that are also hosted on the same server, while the sharing across documents that are hosted elsewhere is not so good.

## 3.3 General Findings and Implications

In general, we found that:

- The sizes of component objects making up a dynamic page can be captured using an *Exponential* distribution. This is in contrast to how static documents are modeled, usually with a heavy-tailed pareto distribution.
- Except for a considerable fraction of objects that change very infrequently, the freshness times of component objects can be captured using a *Weibull* distribution. For two of the sites (`www.amazon.com` and `www.bn.com`), this distribution degenerates into a sharp bimodal one: all of the objects either change on almost every access or change very infrequently.
- Content from all six sites demonstrated significant opportunity for reuse across both temporal and spatial dimensions. More interesting is the relationship between content reuse and object granularity: for four sites (`www.cnn.com`, `dailynews.yahoo.com`, `www.nytimes.com`, `www.windowsmedia.com`), there was a graceful degradation of reusability with increasing object granularity, while the degradation was much sharper for the other two.

Our study of object characteristics of dynamic web content has several implications for future research in optimizing the generation and delivery of dynamic web content:

- An immediate implication, which we explore in the rest of this paper, is that such models enable the development of synthetic content generators, a prerequisite for simulation-based studies of the kind that have previously proved very successful in the static content case.
- Our results also answered most of the questions we posed in Section 1, showing that there is significant opportunity for content reuse, although one needs to distinguish between content that exhibits a graceful degradation of reusability with object size (e.g., `cnn`, `yahoo`, `nytimes`, and `wmedia`) and the kind that exhibits a sharper degradation (e.g., `amazon` and `bn`). That some content exhibits the first kind of behavior is encouraging: it frees up a surrogate or proxy cache designer to work with a range of object sizes in order to achieve benefits from reusability.

[1] The frequent behavior is likely a result of advertisements.

- The Weibull nature of the freshness times distribution suggests that because a large number of objects get updated very frequently, client-initiated cache consistency protocols as opposed to server-initiated ones such as volume leases [29], may be more appropriate for these dynamic objects.

- Knowledge of object size and freshness time distributions can be exploited by a cache designer who can incorporate them into heuristics to decide which objects to cache and which to refetch. For example, based on the caching history, an object can be tagged as being more likely to provide an increased reuse opportunity if it has already been reused a certain number of times.

# 4 DYCE: Dynamic Content Emulator

We designed and implemented a dynamic content emulator (DYCE), which uses the above models to generate parameterizable dynamic content that adheres to the ESI specification. DYCE builds on top of the Tomcat web server from the Jakarta open source initiative [2], and can service requests for both whole documents as well as individual components. DYCE is easy to configure, extend, and use and should prove a useful tool for other researchers in this area. To validate both our models and their use in DYCE, we wrote an idealized cache simulator that can work off both the real trace data as well as the output of DYCE. Comparing the outputs of this simulator for the two cases verifies that DYCE effectively models real content, and at a significant simulation cost advantage. The source code of DYCE is available at `http://www.cs.nyu.edu/~weisong/dyce.html`.

DYCE separates out the functions of content generation and content representation into two modules: the *dynamic content generator* (DCG) and the *dynamic content presentor* (DCP). DCP interfaces with requesters and appears as a traditional web server, capable of servicing two kinds of requests, either for the *document template* or for a set of *selected objects*. The DCG takes three parameters as input to create the corresponding document template and individual objects: the *dynamic content type*, the *size limit* and the *level limit*, two parameters that stem from our proposed splitting schemes.

The DYCE implementation extends the ESI specification in two ways that enables (1) combining of requests for multiple objects from the same document into a single request and response message, and (2) association of per-individual object freshness times. These extensions permit efficient implementation of object-level coherence between a proxy cache or surrogate and a web server.

# 5 Related Work

Web workload characterization has been extensively studied in the past five years from the perspective of proxies [6, 27, 13, 25], client browsers [11, 4, 16], and servers [3, 20]. However, many of these studies focus on the characteristics of web resources at the granularity of the whole document, such as content type, resource size, response size, resource popularity, modification frequency, temporal locality, clent access pattern, and the number of embedded resources. However, for dynamic web content which introduces the notion of object composition, many of these characteristics, such as request and response sizes need to be revisited. Moreover, dynamic content necessitates understanding of new characteristics, such as the number and sizes of objects making up a document, the freshness times of these objects. To the best of our knowledge, the work described in this paper is one of the first efforts trying to model these latter characteristics.

The analysis of web dynamics Brewington and Cybenko [7] and Douglis et al. [13] is very close to our work. However, there are significant differences. First, both these works focus on document-level characteristics, while we are interested in that at the sub-document level. Second, our analysis examines completely dynamic web content, while these works have looked at content with broader characteristics, including a large fraction of static content. Interestingly, the Weibull distribution of freshness time of objects found in our work is similar to that for the expected change time across pages found in [7]. We believe this behavior might arise because of self-similarity characteristics, however, this needs further examination.

Wills and Mikhailov [26] quantitatively analyzed the content reusability present in traces collected from several web sites after a one day interval. Two notable differences of our characterization include the characterization of content reusability at finer granularity (making explicit the time dependence), and the relationship between object size and content reusability. Our results indicate the object granularity that must be supported in order to successfully take advantage of potential reusability.

Challenger et al. [10] analyzed object size distributions based on server traces from the 2000 Olympics site. Although related to our objectives, their method works with a different definition of what an object is: they include not only the individual objects embedded within a document, but also the entire generated document itself. Because of this reason, while a pareto distribution fits their findings, we conclude that the sizes of individual objects in the document follow an exponential distribution.

Our work on the dynamic content emulator resembles work done by Barford et al. on the SURGE static workload generator [5]. However, unlike SURGE, which was designed to model client access patterns, DYCE focuses on the complementary goal of emulating server behavior, both in terms of its load properties as well as the nature of the content itself.

The work in this paper was motivated in part by our inability to extend, to our specific setting, the results previously obtained by researchers working on various aspects of dynamic and personalized content delivery. Such work, which has focused on both server-side [9, 10, 30] and cache-

side [8, 14, 17, 26, 19] has typically been validated with specific, proprietary workloads. For example, Challenger et.al used the 1998 Olympic winter games workload in [9], and the 2000 Olympic games workload in [10], and Douglis et.al used a modified internal AT&T web-based "recruiting database" to evaluate their idea of HPP [14]. The work in this paper complements such real workload studies, more conveniently admitting several "what if" analyses not feasible in the former.

# 6 Conclusions and Future Work

This paper has proposed a methodology for evaluating characteristics of dynamic web content, and used this methodology to obtain models for various independent and derived metrics of interest such as object sizes, freshness times, and content reusabililty. These models have also served as the foundation for the design of a tool, the Dynamic Content Emulator, which emulates a web server serving dynamic content, both in terms of the load properties as well as the nature of the generated content. Our future work consists of using DYCE to evaluate and further refine the design of our CONCA prototype [23], which incorporates a novel design for efficient caching of dynamic and personalized content.

# References


[1] Akamai Technologies Inc. Edgesuite services, `http://www.akamai.com/html/en/sv/edgesuite_over.html`.

[2] Apache Jakarta Project, `http://jakarta.apache.org`.

[3] M. Arlitt and C. Williamson. Internet web servers: Workload characterization and performance implications. *IEEE/ACM Transactions on Networking*, 1997.

[4] P. Barford, A. Bestavros, A. Bradley, and M. E. Crovella. Changes in web client access patterns: Characteristics and caching implications. *World Wide Web, Special Issue on Characterization and Performance Evaluation* 2:15–28, 1999.

[5] P. Barford and M. E. Crovella. Generating representative web workloads for network and server performance evaluation. *Proceedings of Performance '98/ACM SIGMETRICS '98*, July 1998.

[6] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web caching and zipf-like distributions: Evidence and implications. *Proc. of the IEEE Conference on Computer Communications (INFOCOM'99)*, Mar. 1999.

[7] B. E. Brewington and G. Cybenko. How dynamic is the web? *Proc. of the 9th International World Wide Web Conference*, May 2000.

[8] P. Cao, J. Zhang, and K. Beach. Active cache: Caching dynamic contents on the web. *Proc. of IFIP Int'l Conf. Dist. Sys. Platforms and Open Dist. Processing*, 1998.

[9] J. Challenger, A. Iyengar, and P. Dantzig. A scalable system for consistently caching dynamic web data. *Proceedings of Infocom'99*, Mar. 1999.

[10] J. Challenger, A. Iyengar, K. Witting, C. Ferstat, and P. Reed. A publishing system for efficiently creating dynamic web content. *Proc. of the IEEE Conference on Computer Communications (INFOCOM'00)*, Mar. 2000.

[11] M. E. Crovella and A. Bestavros. Self-similarity in world wide web traffic: Evidence and possible causes. *IEEE/ACM Transactions on Networking* 5(6):835–846, 1997.

[12] R. B. D'Agostino and M. A. Stephens, editors. *Goodness-of-Fit Techniques*. Marcel Dekker, Inc, 1986.

[13] F. Douglis, A. Feldmann, B. Krishnamurthy, and J. Mogul. Rate of change and other metrics: a live study of the world wide web. *Proc. of the 1st USENIX Symposium on Internet Technologies and Systems (USITS'97)*, Dec. 1997.

[14] F. Douglis, A. Haro, and M. Rabinovich. HPP:HTML macro-pre-processing to support dynamic document caching. *Proc. of the 1st USENIX Symposium on Internet Technologies and Systems (USITS'97)*, Dec. 1997.

[15] IBM Corp. Websphere platform, `http://www.ibm.com/websphere`.

[16] T. Kelly. Thin-client web access patterns: Measurements for a cache busting proxy. *Proc. of the 6th International Workshop on Web Caching and Content Distribution (WCW'01)*, June 2001.

[17] M. Mikhailov and C. E. Wills. Change and relationship-driven content caching, distribution and assembly. Tech. Rep. WPI-CS-TR-01-03, Computer Science Department, WPI, Mar. 2001.

[18] J. C. Mogul, F. Douglis, a. Feldmann, and B. Krishnamurthy. Potential Benefits of Delta-Encoding and Data Compression for HTTP. *Proc. of the 13th ACM SIGCOMM'97*, Sept. 1997.

[19] A. Myers, J. Chuang, U. Hengartner, Y. Xie, W. Zhang, and H. Zhang. A secure and publisher-centric web caching infrastructure. *Proc. of the IEEE Conference on Computer Communications (INFOCOM'01)*, Apr. 2001.

[20] V. N. Padmanabhan and L. Qiu. The content and access dynamics of a busy web site: Findings and implications. *ACM SIGCOMM'2000*, 2000.

[21] V. Paxson. Empirically-derived analytic models of wide area TCP connections. *IEEE/ACM Transactions on Networking*, 1994.

[22] W. Shi, E. Collins, and V. Karamcheti. Modeling object characteristics of dynamic web content. Tech. Rep. TR2001-822, Computer Science Department, New York University, Nov. 2001, `http://www.cs.nyu.edu/~weisong/papers/tr2001-822.pdf`.

[23] W. Shi and V. Karamcheti. CONCA: An architecture for consistent nomadic content access. *Workshop on Cache, Coherence, and Consistency(WC3'01)*, June 2001.

[24] M. Tsimelzon, B. Weihl, and L. Jacobs. ESI language sepcification 1.0, 2000, `http://www.esi.org`.

[25] D. Wessels. *Web Caching*. O'Reilly Inc., 2001.

[26] C. E. Wills and M. Mikhailov. Studying the impact of more complete server information on web caching. *Proc. of the 5th International Workshop on Web Caching and Content Distribution (WCW'00)*, 2000.

[27] A. Wolman, G. M. Voelker, N. Sharma, N. Cardwell, A. Karlin, and H. M. Levy. On the scale and performance of cooperative web proxy caching. *Proc. of 17th ACM Symposium on Operating Systems Principles (SOSP)*, 1999.

[28] W3C XSL Working Group, `http://www.w3.org/Style/XSL/`.

[29] J. Yin, L. Alvisi, M. Dahlin, and C. Lin. Hierarchical cache consistency in a WAN. *Proc. of the 2nd USENIX Symposium on Internet Technologies and Systems (USITS'99)*, Oct. 1999.

[30] H. Zhu and T. Yang. Class-based cache management for dynamic web content. *Proc. of the IEEE Conference on Computer Communications (INFOCOM'01)*, Apr. 2001.