Chapter 1

Introduction

1.1 Motivation

Introduced a little over a decade ago, the World-Wide Web has transformed much of the world's economy and will continue to do so. It can be compared to the telephone, television, and automobile as a paradigm shift in the way people work, communicate, and play. As the popular interface to what has been called the "information superhighway", the Web purports to provide instantaneous access to vast quantities of information. However, the perception of time is always relative — access to information on today's Web is rarely instantaneous. While performance continues to improve over time from improvements in bandwidth and device latencies, users continue to desire yet additional speed [KPS+99]. Likewise, content providers continue to make greater demands on bandwidth as it increases.

Good interactive response-time has long been known to be essential for user satisfaction and productivity [DT82, Bra86, Roa98]. This is also true for the Web [RBP98, BBK00]. A widely-cited study from Zona Research [Zon99] provides evidence for the "eight second rule" — in electronic commerce, if a Web site takes more than eight seconds to load, the user is much more likely to become frustrated and leave the site. Thus there is also significant economic incentive for many content providers to provide a responsive Web experience.

1.1.1 Methods to improve Web response times

Many factors contribute to a less-than-speedy Web experience, including heterogeneous network connectivity, real-world distances, and congestion in networks or servers due

to unexpected demand. As a result, many researchers (often as entrepreneurs) have considered the problem of improving Web response times. Some want to improve performance by achieving bandwidth increases and response time improvement through bigger "pipes" or alternative communication technologies. Others want to use the existing infrastructure more efficiently. Web caching, along with other forms of data dissemination, has been proposed as a technology that helps reduce network usage and server loads and improve typical response times experienced by the user. When successful, pre-loading Web objects into local caches can be used to further reduce web response times [PM96], and even to shift network loads from peak to non-peak periods [MRGM99]. Our interest is in pre-loading interactively, so that by dynamically pre-loading Web objects likely to be of interest into local caches, we may invisibly improve the user experience by improving the response time.

To be more precise, let us start with a definition:

> *Pre-loading* is the speculative installation of data in a cache in the anticipation that it will be needed in the future. (1)

On the Web, pre-loading typically implies the transmission of data over a network, and can be performed anywhere that a cache is present, including client browsers and caching proxies, but the definition equally applies to cache pre-loading in disk caches, CPU caches, or back-end Web server caches where the source of the data is some other system (such as a database). We use *prefetching* to distinguish the more specific form of pre-loading in which the system holding the cache initiates the retrieval of its own volition. Thus:

> *Prefetching* is the (cache-initiated) speculative retrieval of a resource into a cache in the anticipation that it can be served from cache in the future. (2)

Pre-loading, then, is broader, encompassing both prefetching and *prepushing*, in which content is pushed from server to cache. While this dissertation will focus on prefetching, it will utilize models in which the server sends *hints* specifying the server's recommendation on what should be prefetched.

1.1.2 User action prediction for the Web

Most requests on the Web are made on behalf of human users, and like other human-computer interactions, the actions of the user can be characterized as having identifiable regularities. Much of these patterns of activity, both within a user, and between users, can be identified and exploited by intelligent action prediction mechanisms. Prediction here is different from what data mining approaches do with Web logs. Our user modeling attempts to build a (relatively) concise model of the user so as to be able to dynamically predict the next action(s) that the user will take. Data mining, in contrast, is typically concerned with characterizing the user, finding common attributes of classes of users, and predicting future actions (such as purchases) without the concern for interactivity or immediate benefit (e.g., see the KDD-Cup 2000 competition [BK00]).

Therefore one research focus is to apply machine learning techniques to the problem of user action prediction on the Web. In particular, we wish to be able to predict the next Web page that a user will select. If one were able to build such a user model, a system using it could anticipate each page retrieval and fetch that page ahead of time into a local cache so that the user experiences short response times. In this dissertation, we will demonstrate the use of machine learning models on real-world traces with predictive accuracies of 12-50% or better, depending on the trace.

Naturally, pre-loading objects into a cache is not a new concept, and has already been incorporated into a few proxy caches and into a number of browser extensions (see our Web site on Web caching [Dav02b] for pointers to caching products and browser extensions). Thus we concentrate on the incorporation of a variety of sources of information for prediction, and the principled evaluation and comparison of such systems. We believe that multiple sources are necessary in order to incorporate desired characteristics into the system. Such sources of information would certainly include client history so that an individual's pattern of usage would serve as a strong guide. But we would also want to include community usage patterns (from proxy and origin servers) so that typical usage patterns may be used as intelligent defaults for points in which there is no individual history. Context is also important when history is not relevant

— we use the textual contents of recent pages as a guide to the current interests of the user and the link contents of those pages as significant influences to what may be chosen next. Finally, we recognize that the contents of related applications (such as Usenet news and electronic mail) also present URLs that can be chosen as pages to be retrieved, but leave that as future work to others (e.g., [HT01]). Our conjecture is that the appropriate combination of information from sources such as these will make more accurate predictions possible via a better user model, and thus reduce the amount of extra bandwidth required to generate adequate improvements in response times.

However, the evaluation of such models in terms of response time improvements requires the incorporation of real-world considerations such as network characteristics and content caching. Through simulation, we will model network connections with latencies and bandwidth limitations, thus limiting the transmission speed of content, or delaying its transmission when the simulated network is already at capacity. Embedding prediction mechanisms to perform prefetching within this environment will allow for more accurate evaluation of the efficacy of the prediction techniques. Simulation experiments in this dissertation will show the potential for prefetching on server traces to cut median response times in half or better, and to reduce the sum of all response times by 15-20%, by transmitting an additional 80-85% bytes over what would be transmitted by the equivalent caching-only configuration.

1.1.3 Questions and answers

In general, the quest to pre-load Web content has generated a number of questions:

- How can future Web activity be predicted?
- What methods can be used for pre-loading Web content?
- Can better predictions be made by combining the results of separate prediction systems?

- Does the accuracy of a prediction algorithm reflect its utility in a cache pre-loading environment?
- How can the performance improvements of caching and prefetching techniques be estimated?
- How are implemented caching systems evaluated?
- How can Web cache prefetching systems be evaluated?
- What safety and integrity concerns are there for prefetching systems?
- Why are pre-loading systems relatively uncommon on today's Web?

The answers to these questions, and others, to various degrees, can be found in this dissertation. The search for answers has led to investigation into a variety of areas, including machine learning, simulation, networking, and information retrieval.

What we have found is that while many researchers have proposed various methods for prediction and pre-loading, few are able to accurately evaluate such systems in terms of client-side response times. Generally one may choose from three general approaches to studying system performance: analytic modeling, simulation, and direct measurement. Each provides unique insights into the problem. Analytic approaches provide tools to model systems and scenarios to find trends and limits. Simulation allows for the rapid testing of a variety of algorithms without causing undue harm on the real world. Direct measurements provide grounding in reality with "existence proofs" and challenges for explanations. To realistically consider response times in the Web, however, strict analytic models become unmanageably complex, and thus we have concentrated our efforts on the latter two.

The primary focus of this work has been the design and development of two tools for evaluation of such techniques: a simulator for testing caching and prefetching algorithms, and an evaluation architecture for the testing of implemented proxy caches. Thus, in addition to explorations and surveys of prediction techniques, the majority of the dissertation will propose, implement, validate, and give examples of the use of each tool.

1.2 Contributions

Although there has been significant attention paid to pre-loading from the research community in the time since this thesis was conceived, this dissertation makes a number of contributions around the two themes of this thesis:

- Prediction and Prefetching Methods for the Web

    1. We identify and enumerate key aspects of idealized online learning algorithms.
    2. We propose and demonstrate the utility of a simple approach to user action prediction (realizing an accuracy of approximately 40%) that allows the user's profile to change over time.
    3. We implement a parameterized history-based prediction method, and with it evaluate a space of Markov-like prediction schemes (achieving accuracies of 12-50% or better, depending on the dataset).
    4. We explore the potential for content-based prediction by studying aspects of the Web, and evaluate proposed content-based methods on a full-content Web usage trace. We find a content-based approach to be 29% better than random link selection for prediction, and 40% better than not prefetching in a system with an infinite cache.
    5. We consider the performance improvements possible when combining predictions from multiple sources, and find that combining disparate sources can produce significantly better accuracy than either source alone.
    6. We identify deficiencies in HTTP for prefetching and propose corrective extensions to HTTP.

- Accurate Evaluation

    1. We collect a small, full-content user workload for off-line analysis, since typical usage logs are insufficient.

2. We survey evaluation mechanisms for Web cache pre-loading and reveal problems in them.

3. We identify the need to include network effects in simulations to accurately estimate client-side response times.

4. We implement and validate a new proxy caching and network effects simulator to estimate Web response times at the client. Using it, we find that client prefetching based on Web server predictions can significantly reduce typical Web response times, without excessive bandwidth overhead.

5. We propose and implement a novel black-box proxy evaluation architecture that is specifically capable of evaluating prefetching proxies.

6. We perform experiments using the evaluation architecture to validate the implementation and to test multiple proxies with both artificial and real-world workloads.

1.3 Dissertation Outline

Immediately following this chapter is a primer on Web caching to provide a minimal background and to motivate the rest of the dissertation. The remaining chapters are divided into two parts.

Initially we consider the task of action prediction. The approaches taken for action prediction can vary considerably, and often have complex implementations. In many cases, however, simple methods can do well, and so in Chapter 3 we first consider a relatively unsophisticated approach to the similar problem of using past history to predict user actions within the application domain of UNIX shell commands. We then examine in Chapter 4 the particular problem of prediction of requests for Web objects. Various Markov-like prediction models are detailed, evaluated, and compared under multiple definitions of performance.

These initial chapters also deal with the complementary idea of content-based prediction. Here we consider the contents of Web pages to provide hints and recommendations for future requests. We first explore the potential of content-based mechanisms

by measuring characteristics of the Web, including topical locality and descriptive quality of page proxies in Chapter 5. Having established the potential for content-based prediction, we then consider in Chapter 6 one approach to using the pages that a user visits as a guide to the next page the user will request, and evaluate the approach on a small full-content Web trace.

In later chapters of the dissertation, we deal head-on with more real-world issues, including careful evaluation of caching and prefetching systems. Thus in Chapter 7 we introduce these concerns, and survey proxy cache evaluation techniques. In Chapter 8 we develop and describe an advanced network and cache simulator, and describe efforts to validate this simulator by comparing it to real-world measurements of Web traffic. Chapter 9 then uses the simulator in experiments to incorporate predictions from multiple sources for prefetching. We additionally systematically explore various parameter settings for simulations using two Web server traces to ascertain better performing prefetching configurations.

In contrast to simulation, Chapter 10 introduces a novel architecture for the simultaneous evaluation of multiple *real-world* black-box proxy caches. This architecture is then implemented, tested, and used to evaluate proxy caches in Chapter 11. Chapter 12 discusses the need to change HTTP because of problems with the current HTTP specification and its interpretation. Finally, Chapter 13 wraps up the dissertation by looking ahead to the future of Web caching, and the steps needed to make prefetching widespread.