Journal of Network and Computer Applications (2000) **23**, 201–218
doi:10.1006/jnca.2000.0111, available online at http://www.idealibrary.com on IDEAL®

Performance analysis of a scaleable design for replicating file collections in wide-area networks

Bert J. Dempsey

School of Information and Library Science, University of North Carolina at Chapel Hill, USA. E-mail: dempsey@ils.unc.edu

Mirroring file collections in the global Internet is widely practiced with a recent study estimating the number of WWW hosts with mirrored content at 10% of all WWW hosts. Conventional mirroring tools, however, are not well-suited for the large-scale multiple-site replication services envisioned by projects such as the Internet2 Distributed Storage Infrastructure (I2-DSI) project. This paper presents a scaleable design for the automated synchronization of large collections of files replicated across multiple hosts, as in I2-DSI, and outlines of how the design has been realized using *rsync+*, a modification to the popular open-source mirroring tool, *rsync*. A performance study based on an instrumented mirror using *rsync+* empirically characterizes server-side processing costs under realistic, large-scale workloads, and supplementary measurements of network throughput across Internet2 links illustrate the achievable network performance in a high-speed wide-area network. These experimental results confirm the validity of scalability arguments for the design, uncover key system parameters for *rsync+* that must be tuned for efficient operation, and indicate the limitations of TCP-only transport solutions as the number of mirror sites grows.

1. Introduction

Client-server applications on the Internet benefit from the use of content replication through improved access for widely distributed clients and balancing load across different serves and network paths. Content replication can be employed either in front of the server (proxy-based caching of server responses) or behind the server (mirroring of source objects). On-demand replication of server responses at proxy-based caches is widely deployed within the current WWW where hierarchical or distributed arrangements of cooperating caches handle HTTP requests and, less commonly, those of other access protocols. While an important component of the current and future Internet, proxy-based caching is tied to specific protocols and has well-studied performance limitations. Problematic aspects of proxy caching include delays and bottlenecks caused by hierarchical processing of client requests, the limits of caching dynamically generated server responses, and the tendency of some content providers to eschew caching in order to retain server-side control [1].

Mirroring source objects across server platforms is another approach to content replication that is widely used in the current Internet. A recent study using host

1084–8045/00/030201 + 18 $35.00/0

names extracted from a set of 179 million URLs estimated that about 10% of all WWW hosts had mirrored materials in their filesystems [2]. Among the key advantages of mirroring are (1) the ability to replicate server-side functionality such as server-managed secure logins, (2) replication benefits for any client-server protocol, not just HTTP services, and (3) a basis for constructing manageable, deterministic control of the replication process for the content provider and service provider. In the current Internet, server-side replication is most commonly employed for the limited (but valuable) case of sharing highly popular static file archives served by FTP or HTTP or, less frequently, for more general replication when servers are controlled by a single organization. The latter case implies that strong uniformity can be imposed on the server platforms.

The context of our work is the Internet2 Distributed Storage Initiative (I2-DSI) project [3] which is developing replication middleware to extend source-object replication to loosely-coupled, heterogeneous platforms. I2-DSI relies on a set of dedicated, geographically distributed replication hosts (I2-DSI servers) in the network on which user content will be hosted. Application developers publish collections of source objects (I2-DSI content channels [4]) in conjunction with the I2-DSI interface describing supported content types and the standard services available on the replication hosts. Current content channels are linked at [3]. As part of Internet2, the I2-DSI project aims to develop an open architecture for replication services to serve the 140 Internet2 universities and, subsequently, the wider Internet community. Ideally, over time, standards will emerge for Internet-based replication services, but currently this is an active research area, including the proprietary work being done on application hosting and replication now appearing in commercial efforts [5,6].

In this paper a design is presented for scaleable replication of large collections of files in support of file-oriented replication across the I2-DSI servers.* Widely-used file mirroring solutions in the current Internet automate the process of updating file hierarchies, but they are not well designed to accommodate multiple-site updates in an efficient and timely manner. Our solution for I2-DSI hinges on local generation of the synchronization information (including data from new or updated files) needed for updating a file hierarchy in order to decouple the update process from the transmission of this information across the network. By generating and capturing update information at a master server, the design can eliminate redundant processing at the master server and leverage advances in reliable multicast transports, where available, to distribute the update information efficiently to multiple slave sites.

Asynchronous updates are achieved at the master and slave sites through modifications to a popular state-of-the-art data mirroring tool, *rsync*, and we call our modifications, *rsync+*. *Rsync+* is an application program (written in C) that

* I2-DSI ultimately expects to support multiple replication mechanisms in order to enable the largest possible set of applications to benefit from its replication services.