



Join

Publications ▶

Communities ▶

Digital Library

Get Involved

Member Benefits & Services

Internet Computing home

About Internet Computing

Archives

Search Internet Computing

Author Resources

Media Kit

Subscribe


[HOME](#) [SITE INDEX](#) [SEARCH](#) [HELP](#) [CONTACT](#) [CART](#)

computer.org

Internet Computing
online

Engineering and applying the Internet

[Hot Links](#)[Events](#)[Cache](#)

Guest Editors' Introduction



Scalable Internet Services

Fred Douglass • AT&T Labs-Research

M. Frans Kaashoek • MIT

As the Internet has grown from an experimental system to a commercial reality, the number of users and services has expanded exponentially. When the Internet started, an "Internet service" meant roughly one of three things: electronic mail, file transfer, or remote login. Today, Internet service is nearly synonymous with the World Wide Web, but the Web is not limited to simple hypertext documents -- it includes rich media such as audio and video as well as personalized content in the form of Web portals.

(See the sidebar on [Scalability Resources](#))

Some early commercial uses of the Internet strained its infrastructure, or specific servers on it, beyond anyone's expectations. With 1.5 million hits, for example, the 1999 Victoria's Secret Webcast became a victim of its own success. It is still used as a canonical example of the type of situation to avoid -- a so-called flash crowd that overloads servers or networks and renders them useless. The 1998 distribution of the Starr Report is another example.^[1] The first site to host the report was deluged with requests until other sites that had successfully retrieved the document made their own copies available to spread the load. Though some nascent content distribution networks (CDNs) had already appeared in the marketplace, events like the Starr Report spurred increased public awareness and the entry of new companies such as Akamai.

Managing Uneven Workloads

Scalability issues arise largely because of the gap between average and peak workloads. A service that is provisioned to handle the peak load adequately spends a great deal on resources that will frequently be idle. Conversely, a service that handles the average load well might be swamped by bursts of requests.

Averaging statistical loads across disparate communities is one way to address this problem, as a burst in one community rarely coincides with a burst in another. Hence, CDNs and other large distributed services can gain economies of scale by serving multiple customers, only a small number of which should be susceptible to a flash crowd at once. At the same time, individual server complexes must scale to meet each service's load needs. Portals like Yahoo, for instance, must serve millions of concurrent connections, generating re-sponses that are often customized for each visitor. Thus, services with high amounts of personalized content commonly require a large number of servers, known as a Web server farm, and they use special techniques to make these servers work well together in terms of both performance and accuracy. Scalability within a Web server farm is essential.

Caching and Replication

Caching and replication are two traditional methods for improving performance, and the increasingly blurry division between them falls squarely in the area covered by CDNs. By keeping data close to the user after it has been accessed, caching makes subsequent accesses more efficient. Replication tends to be more proactive, pushing data closer to users or onto additional servers to reduce latency and distribute load.

A common way to replicate data is through mirroring, in which a content owner copies an entire site to multiple locations. CDNs selectively mirror content on servers they control and then direct clients to an appropriate server.^[2] Recently, an increasing number of CDNs are able to generate dynamic data within the CDN, further offloading centralized services.

The Articles

The first theme article, "A Web Caching Primer" by Brian D. Davison, is

intended for audiences with little caching expertise. This tutorial explains how Web caching works, briefly describes the hypertext transfer protocol (HTTP), and discusses different points in the network where data can be cached. It also describes some Web caching problems, such as cache consistency.

In "Lessons from Giant-Scale Services," Eric A. Brewer presents his experiences in designing and planning large-scale industrial Web services. The article addresses issues like load management, high availability, and potential growth of the service. Brewer introduces new availability metrics (yield and harvest) that better capture user experiences than the prior art, and discusses the design implications of these metrics. He also shows one way to bridge the gap between average and peak loads.

If all Web servers dealt only with static data, designing Web services would be easy and many of us "system builders" might be looking for other gainful employment (or at least new research areas). In fact, Web servers often deal with data that changes frequently or at least unpredictably, which makes cache consistency difficult to maintain. Part of the problem is that file systems and Web servers have different notions of cache consistency, and Web servers that share an underlying file system suffer performance penalties because of file system guarantees that are useless when applied to the Web. In our third article, "Efficient Data Distribution in a Web Server Farm," Randal C. Burns et al. describe a new approach to this problem.

We end with "Enhanced Streaming Services in a Content Distribution Network," by Charles D. Cranor et al. Here, the authors report on the design of the Prism architecture, a content distribution network for TV-quality video over IP. Prism allows for different placement, location, and cache replacement policies; it also supports content naming, management, and discovery, as well as content-aware redirection. Finally, the system incorporates network storage and VCR operations, enabling network-based digital VCRs. A preliminary implementation demonstrates the feasibility of the Prism architecture.



References

1. "20 Million Americans See Starr's Report on Internet," CNN.com, 13 Sept. 1998, available at <http://www.cnn.com/TECH/computing/9809/13/internet.starr/>.
 2. B. Krishnamurthy and J. Rexford, *Web Protocols and Practice: HTTP/1.1, Networking Protocols, Caching, and Traffic Measurement*, Addison Wesley Longman, Reading, Mass., 2001.
-

[Fred Douglass](#) is the head of the Distributed Systems Research Department at AT&T Labs-Research. He has a PhD in computer science from the University of California at Berkeley. Douglass is on IEEE Internet Computing's editorial board.

[M. Frans Kaashoek](#) is a professor in the electrical engineering and computer science department at MIT and a member of the Laboratory for Computer Science. He has a PhD from Vrije Universiteit, Amsterdam. Kaashoek's principal interest is designing and building computer systems.

Scalability Resources

Organizations/Standards

- Content Alliance • <http://www.content-peering.org/>
- Content Bridge • <http://www.content-bridge.com/>
- Edge Side Includes • <http://www.edge-delivery.org/>
- Open Pluggable Edge Services • <http://www.ietf-opes.org/>

Newsletters and News Updates

- Stardust CDN Week • <http://www.stardust.com/cdnweek/>
- Web Caching and Content Distribution in the News • <http://www.web-caching.com/news.html>

Top

IC Online is maintained by [Monette Velasco](#)

Need help? Use our [help request form](#)
 Read our [Privacy and Security](#) guidelines.

This site and all contents (unless otherwise noted) are [Copyright](#) ©2004, IEEE, Inc. All rights reserved.

