The following paper was originally published in the
Proceedings of the USENIX Windows NT Workshop
Seattle, Washington, August 1997

High Performance Web Servers on Windows NT Design and Performance

James C. Hu, Irfan Pyarali, and Douglas C. Schmidt
Washington University in St. Louis

For more information about USENIX Association contact:

1. Phone: 510 528-8649
2. FAX: 510 548-5738
3. Email: office@usenix.org
4. WWW URL: http://www.usenix.org

High Performance Web Servers on Windows NT
Design and Performance*

James C. Hu, Irfan Pyarali, and Douglas C. Schmidt
Washington University in St. Louis

Abstract

This research provides two contributions to the study of high-performance Web servers. First, it outlines the optimizations necessary to build efficient and scalable Web servers and illustrates how we applied some of these optimizations to create JAWS, a high-performance Web server that is explicitly designed to alleviate overheads incurred by existing Web servers on high-speed networks. Second, this paper describes how we have customized JAWS to leverage advanced features of Windows NT, such as asynchronous mechanisms for connection establishment and data transfer. Our work includes performance results which characterize the effectiveness of these techniques under increasing server load conditions. We conclude that optimal performance requires adaptive server behavior.

1 JAWS Overview

JAWS is the Web server prototype we developed to analyze Web server performance bottlenecks. Our research involves the empirical study and analysis of the impacts different optimization strategies have to Web server performance as the Web server is subjected to various load conditions, such as request hit rate and requested file size. The strategies under study include: *I/O Strategies*, such as Asynchronous, Synchronous, and Reactive I/O; *Caching Strategies*, such as LRU, LFU, Hinted, and Structured; *Concurrency Strategies*, such as single threaded, thread-per-request, thread-per-session (persistent connections), and thread pool; *Request Handling Strategies*, such as prioritized requests, parallelized protocol processing, and content negotiation; and *Adaptive Protocols*, such as protocol negotiation (PEP), and dynamic protocol pipelines.

2 Performance Results

As shown in [1], JAWS consistently outperforms the other servers in our test suite. These servers included Apache, PHTTPD, Roxen, Netscape Enterprise Server, Zeus, and W^{3}C Jigsaw. During the study, we analyzed the results of our experiments to discover key Web server bottlenecks. We identified the following two key determinants of Web server performance: concurrency and event dispatching strategies; and filesystem access.

Additional experiments described in [2] characterize the relative impacts of different I/O models coupled with different concurrency strategies under various loads on Windows NT over a 155 Mbps ATM network. These experiments revealed two important results. First, throughput is highly sensitive to the I/O strategy and file size. Synchronous I/O mechanisms were found more appropriate for smaller files, while the asynchronous `TransmitFile` appeared most effective for larger files. Second, latency is highly sensitive to hit rate. For smaller files, synchronous I/O provided consistently better performance. For larger files under light loads, `TransmitFile` provided better latency, but is significantly worse than synchronous I/O under heavier loads.

3 Conclusions

These results illustrate that no single Web server configuration is optimal for all circumstances. In order to achieve optimal performance, a Web server must be designed to utilize both *static* adaptivity (bindings of common operations to high performance mechanisms of the native OS) and *dynamic* adaptivity (altering run-time behavior "on-the-fly" based on present load conditions). JAWS provides an application framework which makes this possible on Windows NT.

References

[1] James Hu, Sumedh Mungee, and Douglas C. Schmidt. Principles for Developing and Measuring High-performance Web Servers over ATM. In *Submitted for publication (Washington University Technical Report #WUCS-97-10)*, February 1997.

[2] James Hu, Irfan Pyrarli, and Douglas C. Schmidt. Measuring the Impact of Event Dispatching and Concurrency Models on Web Server Performance Over High-speed Networks. In *Submitted to the 2nd Global Internet Conference*. IEEE, November 1997.

*Additional information on this research is available from the website `http://www.cs.wustl.edu/~jxh/research/`.