

# Handling Persistent Connections in Overloaded Web Servers

Thiemo Voigt  
Swedish Institute of Computer Science\*  
thiemo@sics.se

Per Gunningberg  
Information Technology  
Uppsala University  
Per.Gunningberg@it.uu.se

## 1 Summary

Web servers have to be protected from overload since overload can lead to unpredictable response times, low throughput and even loss of service. To protect servers from overload several admission control architectures have been developed [1, 4, 5, 6, 2, 3, 8].

Many of these architectures can reject or accept connection requests based on information provided in the HTTP header [3, 8]. Persistent connections allow HTTP clients to send several requests on the same TCP connection to reduce client latency and server overhead [7]. Persistent connections are also a challenging problem with respect to admission control since the resource consumption of the requests is unknown at the time the admission control decision has to be made, i.e. when the web server receives the first request on a persistent connection. For example, a user visiting the home page of a company might either leave the company's web pages after visiting a single page or might initiate a long session. Obviously, a long session demands much more resources. Admission control is thus a trade-off. If admission policies are too restrictive, potential customers might be rejected unnecessarily resulting in loss of revenue. If one is too optimistic, the server may become overloaded with unpredictable response times and low throughput as a possible consequence. Even well-engineered adaptive overload prevention schemes suffer from this problem. For example the onset of overload may not be predicted with sufficient accuracy due to workload fluctuations. This defines the goal of our work: To avoid uncontrollable overload while maximizing access.

In [8] we presented kernel-based mechanisms that provide admission control and service differentiation based on filter rules associated with connection and application level information. We have shown that kernel-based mechanisms are more efficient and scalable than controls implemented in user space. In this paper, we extend this architecture to provide kernel-based control of persistent connections. The goal of our overload mechanism is to allow users to complete the sessions (by session, we mean a sequence of individual requests on the same TCP connection) that are regarded as important as well as sessions initiated by users that are regarded as important (for example, users having a service agreement with the site), even when the server becomes overloaded. For example, a session can be regarded as important when the client has placed some items into a shopping bag and thus the likelihood that the client will eventually purchase some items is high. On the other hand, if an unknown user is browsing the site with no evidence that the user might purchase something and an unexpected overload situation occurs, it might be more important to preserve other connections.

We have decided to judge the importance of persistent connections based on the cookies found in the HTTP header. The major advantage of using cookies is that all the information needed to determine the current importance of a connection is found in the HTTP and lower layer protocol headers. Cookie-based

---

\*Thiemo is also at Uppsala University. This work is partially funded by the national Swedish Real-Time Systems research initiative ARTES ([www.artes.uu.se](http://www.artes.uu.se)), supported by the Swedish Foundation for Strategic Research.

connection control is also flexible: When the application is changed, only the filter rules associated with the affected cookies need to be updated. Furthermore, cookies do not only contain information about the current session, but also longer lasting information such as customer identification.

Based around the notion of cookie-based connection control, we extend the architecture presented in [8]. Our experiments, not shown in this summary, show that our approach can prevent server overload and also provide service differentiation under high load. Our approach can also be implemented in the web server or in a middleware layer.

## References

- [1] T. Abdelzaher and N. Bhatti. Web server qos management by adaptive content delivery. In *Int. Workshop on Quality of Service*, June 1999.
- [2] J. Almeida, M. Dabu, A. Manikutty, and P. Cao. Providing differentiated levels of service in web content hosting. In *Proc. of Internet Server Performance Workshop*, March 1999.
- [3] Nina Bhatti and Rich Friedrich. Web server support for tiered services. *IEEE Network*, September 1999.
- [4] L. Cherkasova and P. Phaal. Session based admission control: a mechanism for improving the performance of an overloaded web server. Technical report, Hewlett Packard, 1999.
- [5] V. Kanodia and E. Knightly. Multi-class latency-bounded web servers. In *Intl. Workshop on Quality of Service*, June 2000.
- [6] K. Li and S. Jamin. A measurement-based admission controlled web server. In *Proc. of INFOCOMM*, March 2000.
- [7] J. C. Mogul. The case for persistent-connection http. In *Proc. of SIGCOMM*, 1995.
- [8] Thiemo Voigt, Renu Tewari, Douglas Freimuth, and Ashish Mehra. Kernel mechanisms for service differentiation in overloaded web servers. Usenix Annual Technical Conference 2001.

## 2 More Information

More information on the project can be found at:

<http://www.sics.se/~thiemo/pamp>

A more detailed HTML version of this paper can be found at:

<http://www.sics.se/~thiemo/snart>