**Introduction:**

The current demand for high performance computing is pushing the hardware and software to new limits. As Internet services become more popular and pervasive. More and more content is added online everyday and user s request for information is growing exponentially. A critical problem that arises is managing the performance of services under extreme overload. Building highly concurrent systems, such as large-scale Internet services, requires managing many information flows at once and maintaining peak throughput when demand exceeds resource availability.The real problem is scalability of services.Serve a single user as well as it can serve hundreds or thousands of simultaneous users. The designs of existing operating systems are primarily derived with multiprogramming: allowing multiple applications, each with distinct resource demands, to safely and efficiently share a single set of resources. Internet services have become a vital resource for many people. For example, the Internet based services like e-mail, stock trading and the business web sites are often considered as indispensable. Internet services are a relatively new application domain, which presents unique challenges for OS design. In contrast to the batch processing and interactive workloads for which existing operating systems have been designed, Internet services support a large number of concurrent operations and exhibit enormous variations in load. The number of concurrent sessions and hits per day to Internet sites translates into an even higher number of I/O and network requests, placing great demands on underlying resources. Microsoft’s websites receive over 300 million hits with 4.1 million users a day; Lycos has over 82 million page views and more than a million users daily. Yahoo has over 900 million page views daily. AOL’s Web caches service over 10 billion hits a day.

There are four main reasons that existing OS designs fail to mesh well with the needs of Internet services:

* Inefficient concurrency mechanisms
* Lack of scalable I/O interfaces
* Transparent resource management

**Module Description:**

* Request Handler
* Resource Controllers
* Resource Aware Scheduler
* Thread Pool
* Thread pool performance monitor

**HARDWARE REQUIREMENTS**

Processor : 733 MHz Pentium III Processor

RAM : 128 MB

Hard Drive : 10GB

Monitor : 14” VGA COLOR MONITOR

Mouse : Logitech Serial Mouse

Disk Space : 1 GB

## SOFTWARE REQUIREMENTS

# Platform : JDK 1.6

Operating System : Microsoft Windows NT 4.0 or

Windows 2000or XP

Program Language : JAVA

# Tool : NETBEANS 6.8

**REFERENCES:**

[1]D. Li, J. Wu, Y. Cui, and J. Liu, “QoS-aware streaming in overlay multicast considering the selfishness in construction action," in Proc. IEEE INFOCOM, 2007.

[2] G. Tan and S. Jarvis, “Improving the fault resilience of overlay multicast for media streaming," IEEE Trans. Parallel Distributed Syst., vol. 18, no. 6, pp. 721-734, 2007.

[3] A. T. S. Chan, J. Cao, and C. K. Chan, “Webgop: collaborative web services based on graph-oriented programming," IEEE Trans. Syst., Man, Cybernetics, Part A, vol. 35, no. 6, pp. 1874-1885, 2005.

[4] J. Liu, J. Xu, and X. Chu, “Fine-grained scalable video caching for heterogeneous clients," IEEE Trans. Multimedia, vol. 8, no. 5, pp. 1011- 1020, 2006.

[5] K. Shen, H. Tang, T. Yang, and C. L, “Integrated resource management for cluster-based Internet services," in Proc. OSDI, 2002.

[6] C. Tang, M. Steinder, M. Spreitzer, and G. Pacici, “A scalable application placement controller for enterprise data centers," in Proc. International World Wide Web Conf. (WWW), 2007.

[7] H. Shachnai and T. Tamir, “On two class-constrained versions of the multiple knapsack problem," Algorithmica, vol. 29, no. 3, pp. 442-467, 2001.

[8] ——, “Polynomial time approximation schemes for class-constrained packing problems," J. Scheduling, vol. 4, no. 6, pp. 313-338, 2001.

[9] S. Zhou, “A trace-driven simulation study of dynamic load balancing," Univ. California, Berkeley, Tech. Rep. UCB/CSD87/305, 1986.

[10] A. Karve, T. Kimbrel, G. Pacici, M. Spreitzer, M. Steinder, M. Sviridenko, and A. Tantawi, “Dynamic application placement for clustered web applications," in Proc. International World Wide Web Conf. (WWW),

2006.