Current Research Trends in Internet Servers

K. Kant
Intel Corporation
MS JF1-231, 25111 NE 25th Avenue
Hillsboro, OR 97124
Email: krishna.kant@intel.com

Prasant Mohapatra
Department of Computer Science and Engineering
Michigan State University
East Lansing, MI 48824
Email: prasant@cse.msu.edu

1 Introduction

The article "Scalable Internet Servers: Issues and Challenges" published in PAWS-2000 proceedings [1] identified eleven different areas in Internet Servers that require significant research efforts (available at `http://kkant.ccwebhost.com/PAWS2000`). The main motivation was to spur more interest and research into the Internet server. In this section we add to this list by mentioning a number of research issues on the two new topics encompassed by PAWS-2001, namely peer to peer computing and wireless Internet access. On the former topic, we also describe some of our recent work. We also revisit some of the issues mentioned in [1] and briefly describe some of the work that we have accomplished in these area.

2 Peer to Peer Networking

Popularized by Napster and Gnutella file sharing solutions, peer-to-peer (P2P) computing has suddenly emerged at the forefront of Internet computing. The basic notion of co-operative computing and resource sharing has been around for quite some time, although these new applications have opened up possibilities of very flexible web-based information sharing. (This issue was mentioned in [1] in item 5). A number of companies have advocated peer-to-peer solutions to problems such as distribution of streaming media, web hosting, distributed auctions, etc. Meanwhile there is a renewed interest in a large body of distributed system's research on resource sharing and collaboration in both LAN and WAN environments. In particular, "WAN-OS" projects such as Legion (`www.cs.virginia.edu/~legion`) and Globe `www.cs.vu.nl/~steen/globe/`) are well suited for supporting arbitrary P2P applications since their goal is to make the Internet look like a single parallel machine by hiding (to the extent desired by the programmer) all the complexities associated with vastly different machines, local operating systems, communication protocols, local resource management, access control, and security policies. Recently, major players such as Microsoft and Sun, have announced new initiatives to support complex P2P applications in their respective operating system environments. In the former case, P2P computing is intended as a part of .NET strategy (`www.microsoft.com/net`), which envisions arbitrary services to be deployed over the web via the SOAP interface (`www.w3.org/TR/2001/WD-soap12-20010709/`). In contrast, the JXTA open source project from Sun (`www.jxta.org`) enables P2P applications by specifying a set of protocols for peers to interact with one another.

Although there is no widely-accepted definition of P2P networking, we believe that it should include three major aspects (a) distributed data and/or metadata, (b) incomplete global knowledge, and (c) no strict client-server relationship. An overview of P2P applications may be found in [4], which attempts to provide a taxonomy of P2P applications based on five dimensions: resource (data) location, control (metadata) location, resource usage, consistency constraints, and QoS constraints. The taxonomy also considers environmental attributes such as latency, security, connectivity, etc. in order to address implementation issues for P2P applications. The taxonomy points to a number of research issues that need to be examined in order to attain the full potential of P2P computing. Briefly, the most important issues include (a) devising efficient mechanisms for information location, (b) coping with network address translation and firewalls in providing interaction between peers, (c) intelligent searching and search response propagation mechanisms, (d) hybrid client-server and P2P approaches that can exploit the vast idle resources of P2P environment and yet provide the responsive and reliability of traditional client-server paradigm, (e) lightweight and nimble protocols to ensure good service to both the host and guest (i.e., P2P) applications on peers, (f) coping with various facets of hostile environment for transactional and real-time applications, and (g) performance characterization of P2P computing environment to enable comparative evaluation of many design choices. Reference [5] provides a simple performance model to characterize evolving P2P approaches for the file sharing class of applications; however, much further work remains to be done on this topic.

3 Wireless Internet Services

Wireless mobile Internet access has been an area of great interest over the past several years and simple services (e.g., limited web-browsing, display of local facilities and attractions, etc.) are already being provided. Given the promise of 3G and 4G wireless technology with substantial bandwidths and more capable end devices, richer services should be feasible. However, there are several constraints in the wireless mobile environment that cannot be removed easily and perhaps will stay on as echoed by the PAWS-2001 panel session as well. These are: (a) the need to support a wide variety of devices with a large range of capabilities, (b) very limited power budget for the device, and (c) limited scope for interaction by a mobile user. These characteristics have important implications for server design for mobile wireless environment. In particular, the servers must be able to efficiently provide content matched to device's capabilities. As wireless Internet usage becomes more ubiquitous, it will also be necessary to allow transparent switching between multiple devices depending on the user convenience and current network conditions (static vs. moving, battery powered vs. live power source, crossover into a different network, etc.). The dynamic switchability could significantly complicate content filtering needed to support various types of devices.

Limited power budget requires that the server do most of the work and thus requires asymmetric protocols that concentrate processing burden on the server. The limited scope for interaction for a mobile user implies that servers need to support much more sophisticated interactions than simple browsing such that the desired information can be easily asked for and delivered. In particular, a voice recognition based natural language query mechanism may be appropriate for a driver, but URL requests or elaborate keypad/screen based queries are not. Because of the power limitations on the client side, the natural language queries would have to be sent directly to the gateway for interpretation and response generation. The desired response is no longer a web-page, but an answer crafted by perhaps obtaining information from multiple end servers and manipulating this information in complex ways (perhaps followed by response delivery via synthesized voice). Currently, complex decision making in the process of response generation belongs primarily to the domain of business-to-business (B2B) e-commerce. Adapting this capability to the general wireless web user environment brings in enormous challenges in terms of scalability of providing wireless Internet access, particularly when coupled with the need to support a large variety of devices in a dynamic environment and intelligent processing of queries.

4 Overload Control in Web Servers

Item 9 in [1] indicated the need for good overload control schemes in web servers. This is particularly important since web servers often experience overload situations due to the extremely bursty nature of Internet traffic, popular online events, or malicious attacks. Such overload situations significantly affect performance and may result in lost revenue as reported by the recent denial of service attacks. Reference [2] studies three simple schemes for controlling the load effectively. The first scheme selectively drops incoming requests as they arrive at the server using an intelligent network interface card (NIC). The second scheme provides feedback to a previous node (proxy server or the ultimate client) to allow a gapping control that reduces offered traffic under overload. The third scheme is simply a combination of the two. The experimental results show that even these simple schemes are effective in improving the throughput of the web server by 40% and response time by 70% under heavy overloads, as compared with the case where no overload control is effected. The paper also addresses the issue of overload control when the traffic involves multiple classes of traffic, each with a different quality of service requirement. One example of such an environment is a server handling both secure and non-secure HTTP requests. It is shown in [3] that processing of secure HTTP requests via the secure sockets layer (SSL) protocol involves very substantial increase in processing requirements and substantial change in the workload characteristics. Yet, secure transactions are perhaps more important in a business-to-consumer (B2C) e-commerce environment as they are often associated with revenue generation. A differentiated overload control using the concept of congestion priorities could provide the required support in this case. The current overload control scheme is intended only for short duration connections since it does not deal with individual requests going over the connection. The paper by Voigt and Gunningburg in this issue is relevant in this regard as it deals with a cookie-based scheme for server overload protection under persistent connections.

In a recent work [9], we have further explored a session-based overload control technique. A session-based overload control technique would be more meaningful in e-commerce environment compared to the request-based techniques. In these enviornment, the number of completed sessions relate to the number of transaction completions (which may be more meaningful from a revenue-generation standpoint). A probabilistic model for the state transitions within a session is first derived in this work. A dynamic weighted fair scheduling approach is employed which helps in avoiding the processing of *unproductive* requests (the requests that belong to a session that is likely to get aborted). Thus, the overload is controlled because of the processing of only those requests that are likely to contribute to the completion of a session.

Furthermore, issues such as service differentiating Internet servers [6] and admission control [7] can be extended to handle overload control based on quality of service (QoS) requirements.

5 Coping with Dynamic Content

Item 2 in [1] noted the challenges in coping with the increase in dynamic content in the world wide web. In the e-commerce environment these objects form the core of all web transactions. However, because of additional resource requirements and the changing nature of these objects, the performance of accessing dynamic web contents has been observed to be poor in the current generation web services. We have proposed a framework called *WebGraph* that helps in improving the response time for accessing dynamic objects [8]. The WebGraph framework manages a graph for each of the web pages. The nodes of the graph represent *weblets*, which are components of the web pages that either stay static or change simultaneously. The edges of the graph define the inclusiveness of the weblets. Both the nodes and the edges have attributes that are used in managing the web pages. Instead of recomputing and recreating the entire page, the node and edge attributes are used to update a subset of the weblets are are then integrated to form the entire page. In addition to the performance benefits in terms of lower response time, the WebGraph framework facilitates web caching, QoS support, load balancing, overload control, personalized services, and security for both dynamic as well as static web pages. We have implemented the WebGraph framework in an experimental set-up and have measured the performance improvement in terms of server response time, throughput, and connection rate. The results demonstrate the feasibility and validates a subset of the advantages of the proposed framework.

6 Conclusions

In addition to the issues discussed here, several other issues, such as adaptability of web services, QoS-aware web servers, and the content delivery edge services, are also emerging as important and interesting research topics. We would like to receive feedback from the readers on all the topics discussed in this paper.

References

[1] K. Kant and P. Mohapatra, "Scalable Internet Servers: Issues and Challenges", PAWS-2000 proceedings, Aug 2000.

[2] R. Iyer, V. Tewari, and K. Kant, "Overload Control Mechanisms for Web Servers", Performance and QoS of Next Generation Networks, Nagoya, Japan, Nov 2000, pp 225-244

[3] K. Kant, R. Iyer and P. Mohapatra, "Architectural Impact of Secure Socket Layer on Internet Servers", International Conference on Computer Design (ICCD 2000), Sept 2000, pp 7-14.

[4] K. Kant, R. Iyer and V. Tewari, "On the potential of peer-to-peer computing: Classification and Evaluation", `http://kkant.ccwebhost.com/download.htm`

[5] K. Kant, R. Iyer and V. Tewari, "A performance model for peer to peer file-sharing services", `http://kkant.ccwebhost.com/download.htm`

[6] X. Chen and P. Mohapatra, "Providing Differentiated Service from an Internet Server," Int. Conference on Computer Communications and Networks, pp. 214-217, 1999.

[7] X. Chen, H. Chen, and P. Mohapatra, "An Admission Control Scheme for Predictable Server Response Time for Web Accesses," Proc. of the International World Wide Web Conference, pp. 545-554, May 2001.

[8] P. Mohapatra and H. Chen, "WebGraph: A Framework for Managing Dynamic Web Contents," `http://www.cse.msu.edu/ prasant/`

[9] H. Chen and P. Mohapatra, "Session-Based Overload Control in Web Servers," `http://www.cse.msu.edu/ prasant/`