

# A Non-interfering Deployable Web Prefetching System

Ravi Kokku, Praveen Yalagandula,  
Arun Venkataramani, Mike Dahlin

USENIX Symposium on Internet Technologies &  
Systems (USITS), March 2003



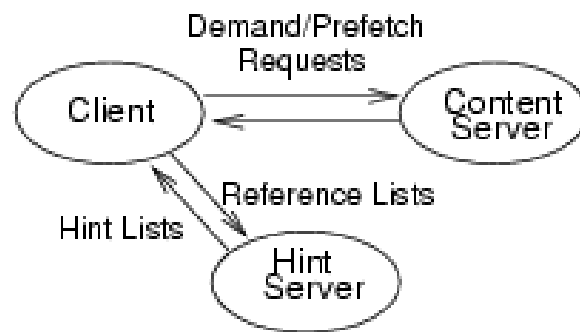
Presented by Prof. Davison  
24 April 2003

# Prefetch-Nice

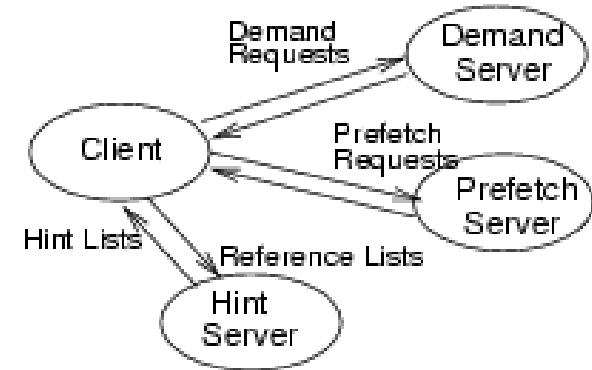
- Web prefetching system
- Claims:
  - Novel
  - Non-intrusive
  - Avoids interference between prefetch and demand requests at server and network
  - Deployable without modifications to browsers, HTTP protocol, and network
  - 25% reduction in response time vs. traditional technique

# How Prefetch-Nice works

- To prefetch, it
  - modifies HTML pages to include JavaScript code to issue prefetch requests
- To avoid interference, it
  - monitors the server load externally to dynamically tune client prefetch aggressiveness
  - uses TCP-Nice, a specially tuned TCP stack for low-priority data transfer
  - uses heuristics to control resources on the client



(a) One Connection



(b) Two Connection

Figure 1. Design Alternatives for a Prefetching System

# Architectures

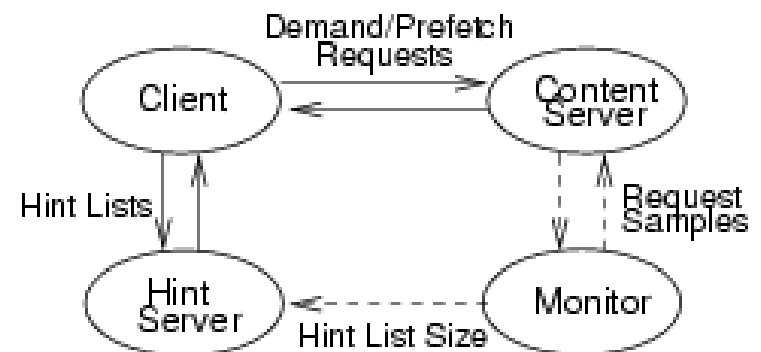
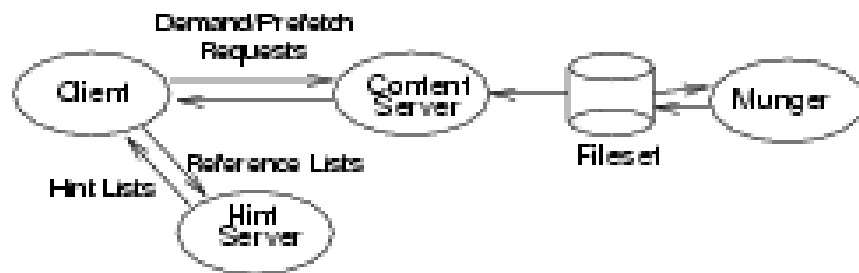
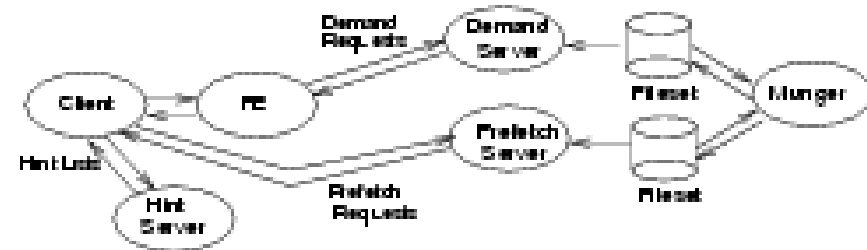


Figure 3. A Monitored Prefetching System



(a)

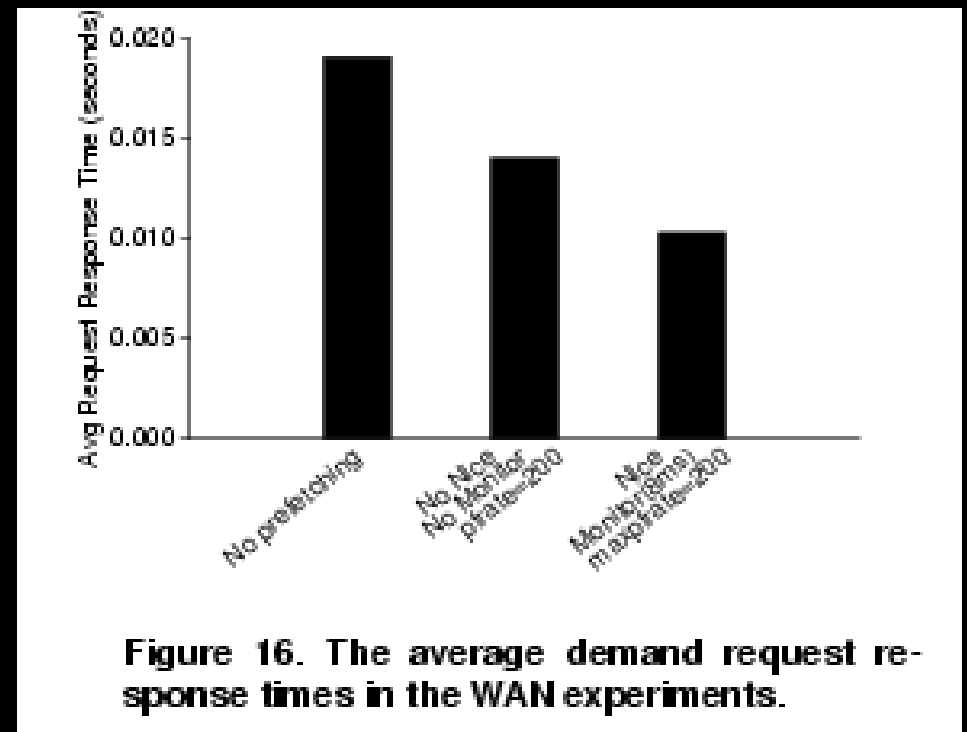
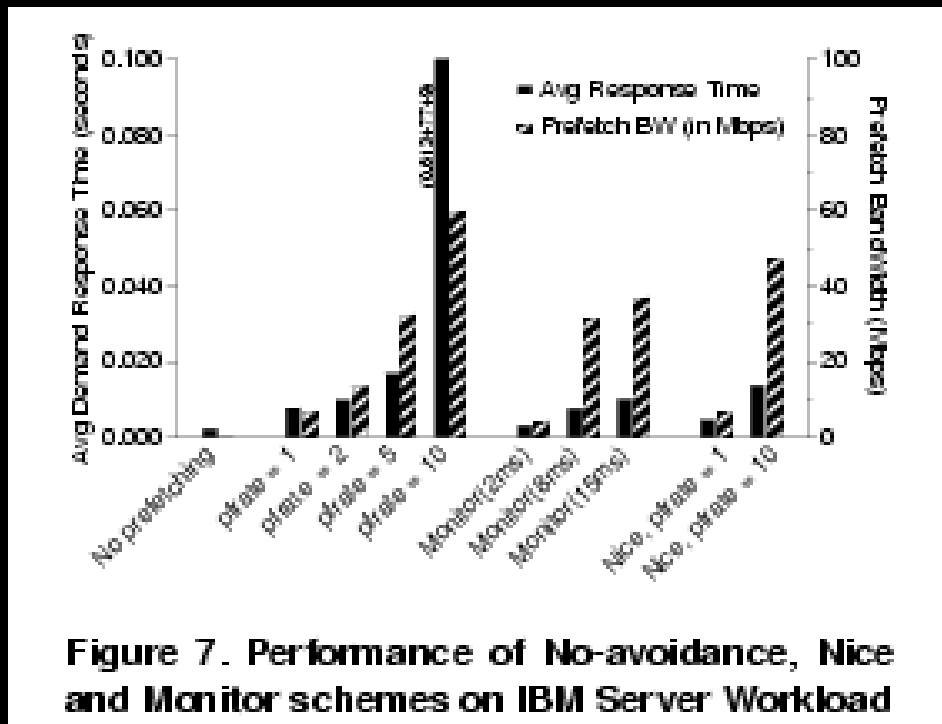


(b)

Figure 10. Prefetching mechanisms for (a) one connection and (b) two connection architectures.

# Results

- IBM sporting event workloads on LAN and WAN (cable modem) environments



# Critique

- Paper had good goals
- Did not succeed – requires changes to server (TCP-Nice), requires infrastructure (monitor, page munger, prefetch server), prefetching technique not new, requires separate hint retrievals and reference uploads, prefetched objects have different URLs, monitor causes overhead
- Experimental results are very limited (one server-only dataset, two environments), and not always positive (all prefetching increased mean response time in fig 7, and TCP-Nice paper had similar data)
- Paper unclear – how to run JavaScript-based experiments? Monitor(?ms)? Placement of hint server?