Supporting Context-Aware Applications Using Thin Clients

Cynthia Taylor, Joe Pasquale
University of California, San Diego

Introduction

•This project explores a new systems architecture and programming paradigm for Thin Client computing.
•In this architecture, a Smart Proxy is added to the standard client-server system.
•The Smart Proxy can be added to an unmodified client-server system, and can be used for either modification of updates or buffering/caching of updates.

Devices

Zypad Wearable Computer

Video Glasses

Chumby

iPhone

Cell Phone

The current proliferation of cheap, lightweight devices offers a perfect opportunity for the use of Thin Client systems.

Applications

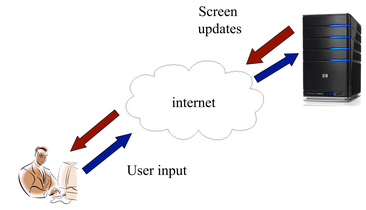
Virtual Worlds

Maps

Historic Information

Since we carry these devices with us everywhere, it would be ideal if they knew about our location and could give us contextual information about our surroundings.

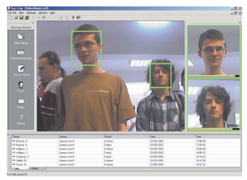
What is Thin Client Computing?



In Thin Client computing, all applications run on the server, and the client is used predominantly for I/O. User input is sent to the server, and screen updates are sent to the client.

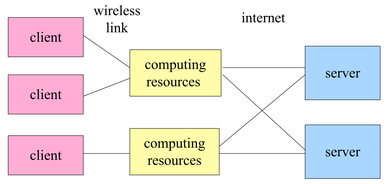
Why Thin Clients? Intensive Applications

- Machine Learning/Vision
  - Object recognition
  - Speech recognition
- Graphics
  - Rendering
- Data Storage
  - Video



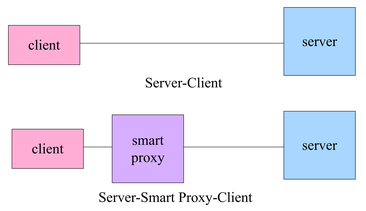
The applications that fully exploit knowledge of our surroundings have high processing and storage needs.

Resource Assumptions: Active Wireless Spaces



Our proposed system architecture assumes that there are available computational resources attached to WiFi routers, provided in the same way an office, university, or coffee shop will currently provide wireless internet.

Server-Client vs Server-Smart Proxy-Client



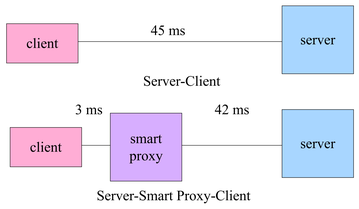
The Smart Proxy runs on these available computing resources. Because it can communicate with the client very quickly, we can move tasks that would be done on the client machine to the Smart Proxy.

Uses of the Smart Proxy

•Buffering updates
•Caching Location-Dependent Information
•Compress or Decompress Updates
•Scalable Video Coding
•Video Processing
•Encryption

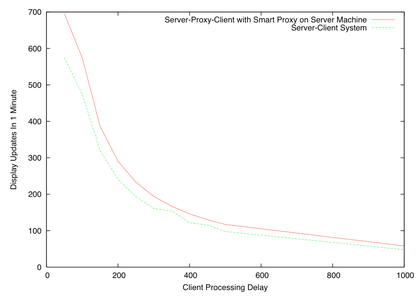
The Smart Proxy is ideal for any task that requires frequent quick communication with the client, or any task that requires specialized knowledge of its location, since it is stationary.

Adding a Smart Proxy to VNC



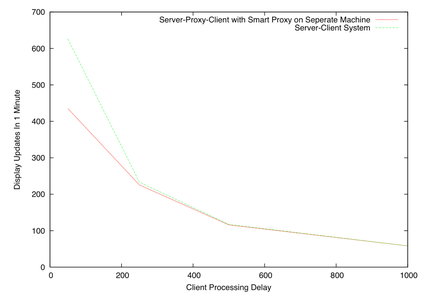
Having the Smart Proxy continuously query the server for display updates means a display update will be ready at the Smart Proxy for it to immediately send to the client, taking the client 3 ms to receive an update instead of 45 ms.

Preliminary Experimental Results



When the Smart Proxy is run on the same machine as the VNC server, the system is able to supply significantly more screen updates to the client. This is because the updates have already been generated in the proxy system, and the proxy is optimized to send messages more efficiently.

Preliminary Experimental Results



When the Smart Proxy is run on a separate machine, it delivers fewer updates than the unmodified system. We believe this is due to per-packet and per-message overheads in VNC, which we are working to mitigate.

Conclusion

•Smart Proxies offer a way to perform computationally difficult tasks with quick update speeds, while requiring the user to carry only light-weight, low-power devices.

•Using this systems architectures lets users use devices they already carry with them for location/context-dependent, computationally-intensive applications.

•Initial experiments on with VNC show that the Smart Proxy system has the potential for performance advantages in existing client-server systems, without modifications to the existing code.