Collaborative Client-Server Architectures in the Web-based Viewing Scheme

Peter M. C. Chen
Philips Research East Asia – Taipei (PREA-T), Philips Corporation
Taipei, Taiwan, R.O.C.
+886-2-23823049
peter.mc.chen@philips.com

Wei-Guang Teng
Electrical Engineering Department, National Taiwan University
Taipei, Taiwan, R.O.C.
+886-2-23635251
eev@arbor.ee.ntu.edu.tw

ABSTRACT

In this paper, a technique for enhancing viewing capabilities of web-based clients, called the Super Plug-in Module (SPM), is proposed. By utilizing the collaborative client-server architecture, clients can view various types of documents through the help of an application server. The application server performs document transcoding operations while acts as a proxy at the same time, thus alleviating loads of clients. And it is not necessary to install additional software modules on clients equipped with the SPM when new document types are introduced. This feature results in great benefits for the SPM technique over typical ones. Two alternative scenarios simulating different real situations are constructed. Experimental studies conducted in these two scenarios show promising results of this work.

Keywords

collaborative client-server architecture, thin client, document transcoder, plug-in technique

1. INTRODUCTION

Viewing unknown types of documents, an appropriate plug-in module has to be installed and executed on the web-based clients for decoding purposes. However, the consumption of system resources resulted from these plug-in modules could post a heavy burden to so-called resource-constrained devices. In this paper, we propose a technique, called Super Plug-in Module (SPM), and the corresponding architecture to alleviate the problems mentioned above. By employing the SPM and the help from a relaying computer whose role is the document transcoder, resource-constrained devices can thus view various types of web-based document objects without the threaten of being overloaded.

The thin client, which was originated from the concept of X Windows, has become a general term for a simple and low cost device with abilities to access centralized server applications in the pervasive computing world. In essence, the thin-client/server computing [1] is a computation model due to the need of reducing the Total Cost of Ownership (TCO) at work. Currently, this model is well suitable for office applications such as spreadsheets.

In the collaborative client-server computing model proposed in this paper, the task of web-based viewing is split into the client part and the server part. The client part is a primitive web browser with a plug-in module, i.e., the SPM-client. The server part, which executed in application server, can be further divided into the SPM-server and the SPM-transcoder.

Whenever the primitive web browser finds a new media type, an SPM-client instance is created. This SPM-client instance knows merely where the actual decoder, i.e. the application server, is. Then it can delegate the decoding work to that application server via passing required information, for example URL, to the SPM-server.

The SPM-server further dispatches the document transformation request to the SPM-transcoder. The SPM-transcoder then can generate some known output formats that the SPM-clients or the primitive web browsers can decode and display on client device. Figure 1.1 depicts the overall SPM architecture and the relationship among key components.

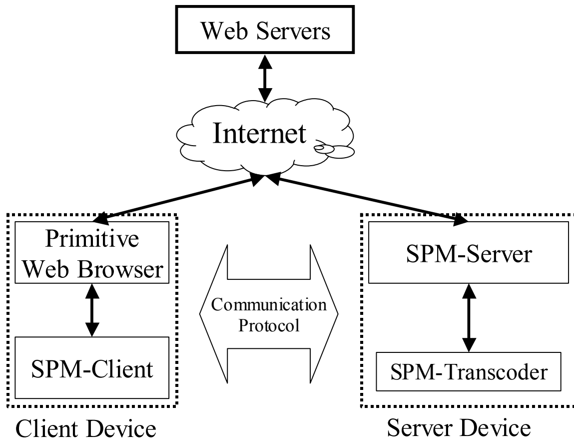


Figure 1.1 SPM system architecture and key components

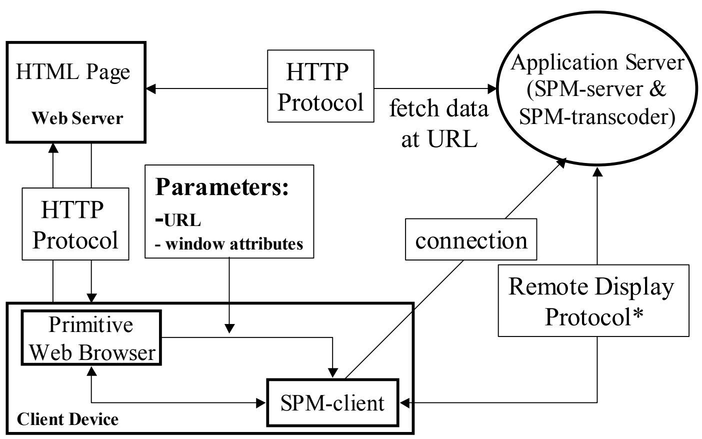
2. FRAMEWORK OF THE SPM

In this section, two approaches of the SPM technique are investigated. The first approach, called screen-capture, is based on the concept of the remote display protocol [2]. The second one, called virtual-printer, is based on the concept of the document transcoder.

The platform of SPM-clients could be implemented as a plug-in of Netscape or an ActiveX control of Internet Explorer. It is registered to the primitive web browser as is capable to decode various document types. Both of our client and server platforms currently are based on PC Windows's platforms without modifying any underlying operating systems.

2.1 Screen-Capture Approach

Whenever a document hyperlink is clicked and MIME type is recognized by the SPM-client, it makes a connection with the SPM-server. By leveraging the Active Document Host feature of underlying Windows platform, the SPM-transcoder can capture the updated screen data and then encode it to some known format that the SPM-client to decode and display. Figure 2.1 demonstrates the architecture framework of screen-capture approach.



*: Propriety peer-to-peer communication protocol

Figure 2.1: Architecture view of screen-capture approach

The SPM-client only deals with some I/O interactive functions like mouse click, or screen update events. The connection between the SPM-client and the SPM-server will be terminated whenever the user leaves the viewing web page. As of remote display characteristics, this approach possesses same network bandwidth consumption characteristics as Virtual Network Computing [3].

2.2 Virtual-Printer Approach

As we know there is a well-defined interface, called Graphics Device Interface, between applications and printer drivers of underlying Windows platform. Thus allow us capable of hooking and filtering print commands before sending to target device. As such, the SPM-transcoder is implemented as virtual printer driver. Each time the SPM-server needs to perform document transformation, it defers the requests to the SPM-transcoder for decoding. Figure 2.2 demonstrates the architecture framework of virtual-printer approach.

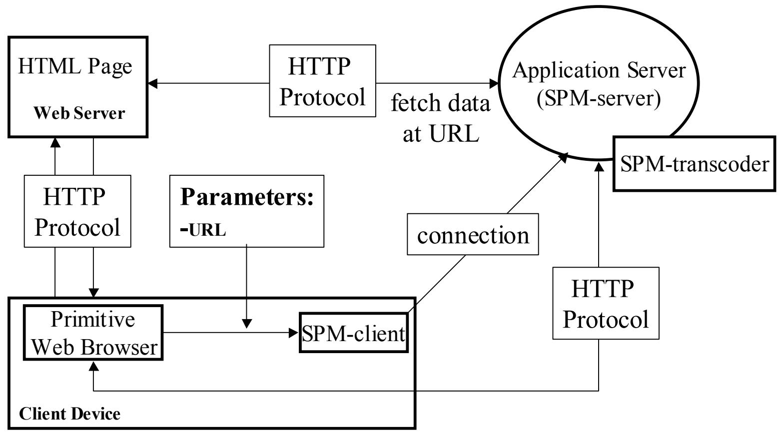


Figure 2.2: Architecture view of virtual-printer approach

This approach utilizes the rendering interfaces of printer components of underlying Windows platform. So the SPM-transcoder can automatically transform the document file into sequences of viewing pages, which could be in some standard format such as JPEG. In addition, this approach can support concurrent multiple-session connections without using expensive server.

3. EMPIRICAL RESULTS

In order to evaluate the performance of our SPM architectures, two different scenarios: the local proxy and the server extension, reflecting the possible relationships among the web server, the web-based client, and the application server are assessed.

In the local proxy scenario, the application server acts as a proxy server near end-users' client devices. It helps the client devices on retrieving documents locating at some web server and then transcodes the documents according to clients' requests. In the server extension scenario, the application server acts as an extensive server in addition to the original web server.

3.1 Benchmark Environment

Here we propose a benchmark to assess the solutions with the ordinary approach, i.e. local plug-in solution of commercial full-feature web browser, from the following respects: document size, network architecture, and response time performance. To be in a controllable way, our test platform is based on closed LAN environment with WAN delay router emulator. Figure 3.1 shows the configuration of benchmark.

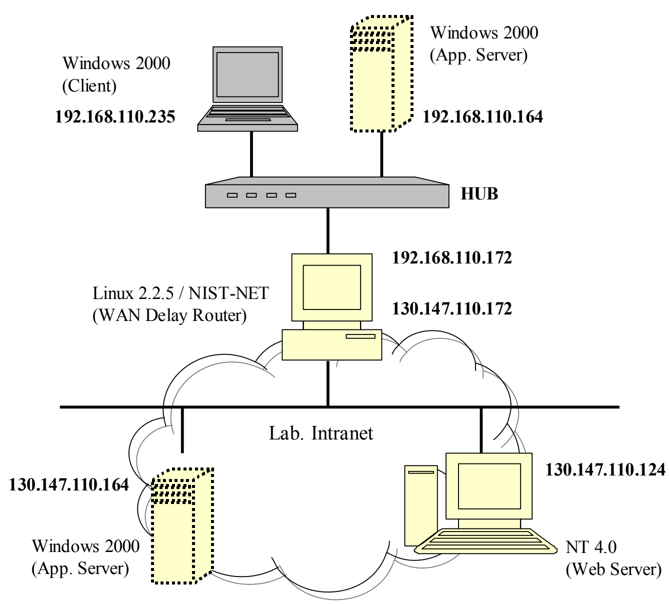


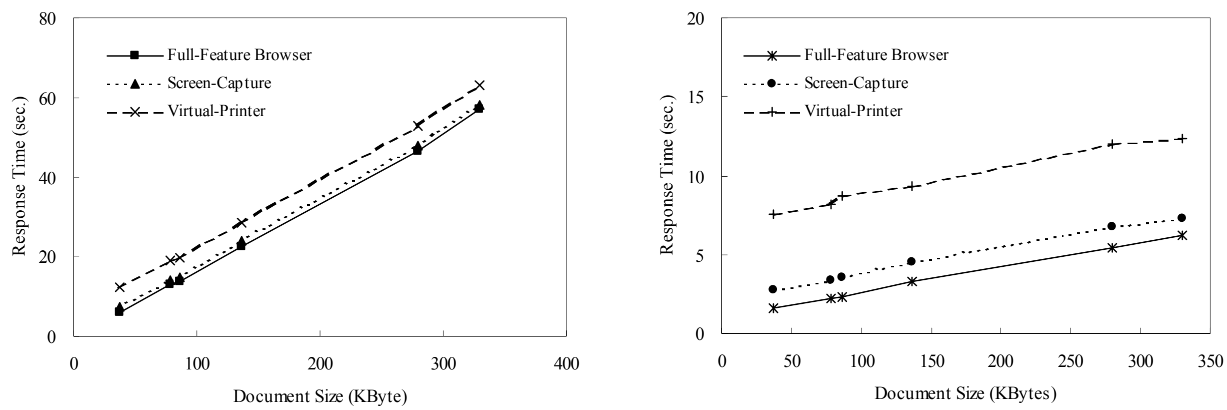
Figure 3.1: Network configuration for performance benchmark

Recent research [4] shows the long-range dependence (LRD) is a ubiquitous property of traffic for both WAN and LAN. To facilitate our experiments in a networking environment, the "NIST-NET (Network Emulation Tool by National Institute of Standards and Technology)" is adopted. This software emulator is operated at the IP layer and provides supports on Pareto distribution packet delaying.

Two pathways, which are commonly used by home users, are designated for our performance tests. One is the modem connection with a 56Kb/s bandwidth, and the other is the ADSL connection with a 512Kb/s downlink and a 64Kb/s uplink bandwidths.

3.2 Performance Studies

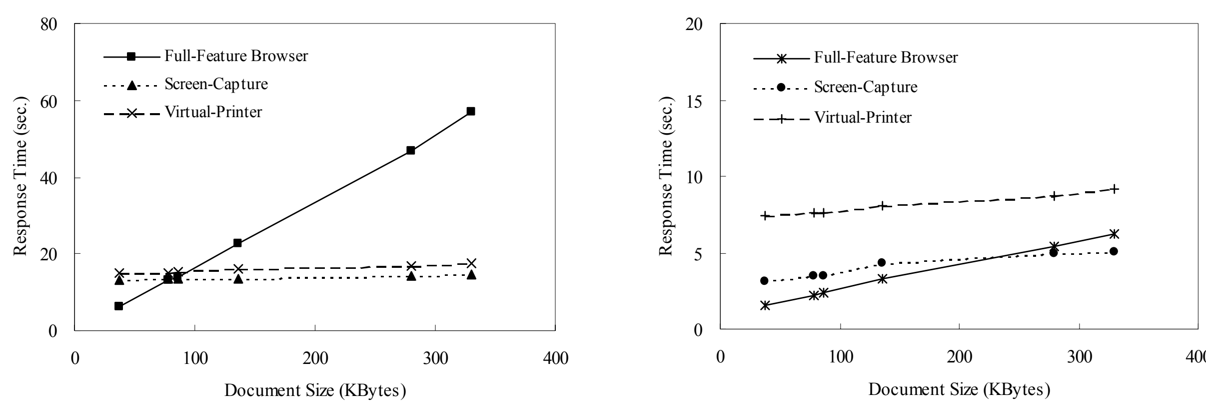
Figure 3.2 shows the response time with respect to the document file size for local proxy scenario. The response time makes no difference when network bandwidth between the client and the web server is low. This is because network bandwidth dominates the response time. The nearly constant overhead between screen-capture and virtual-printer introduced is the results of processing overhead on application server.



(A) 56K modem connection **(B) 512K/64K ADSL connection**

Figure 3.2: Performance results in the local proxy scenario

Figure 3.3 shows the response time with respect to the document file size for server extension scenario. Both of the SPM architectures improve the response time performance for server extension scenario. Especially the screen-capture approach gets even more improvement under low network bandwidth between the client and the web server since the transformed documents are of smaller sizes which saves transmission time greatly.



(A) 56K modem connection **(B) 512K/64K ADSL connection**

Figure 3.3: Performance results in the server extension scenario

4. CONCLUSIONS

In this paper, the SPM technique and two corresponding architectures, i.e. screen-capture and virtual-printer, are proposed in solving the problem of viewing documents with resource-constrained devices. The resource-constrained devices can leverage the support from an application server to extend the capability of viewing document on the web. Extensive experimental studies show promising results of performance improvement through the introduction of the SPM architecture. And in the server proxy scenario, the SPM architecture gets much more effective response time improvement. In addition, the virtual-printer approach could support concurrent multiple-session connections without relying on server host, which is usually powerful and thus expensive. For resource-constrained devices, the resource costs are proved to be reduced to minimum as expected.

5. ACKNOWLEDGEMENTS

The authors would like to thank supports from PREA-T Department Head Marcel Fuhren, and Janet J.C. Chen.

6. REFERENCES

1. Joel P. Kanter, "Understanding Thin-Client/Server Computing," Microsoft Press, 1998.
2. George Hadjiyiannis, Anantha Chandrakasan and Srinivas Devadas, "A low power, low bandwidth protocol for remote wireless," Wireless Networks 4, 1998, pp. 3-15.
3. Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood and Andy Hopper, "Virtual Network Computing," IEEE Internet Computing, Vol. 2, No. 1, January/February 1998, pp. 33-38.
4. Qiong Li and David L. Mills, "On the Long-range Dependence of Packet Round-trip Delays in Internet," Proceedings, IEEE ICC '98, pp. 1185-1191, Aug. 1998.