An Application Sharing Platform with Low Management Costs

Masafumi Higuchi[†]*, Manabu Okamoto, Naoaki Shinoda, and Naohisa Tanabe*

Abstract

NTT Information Sharing Platform Laboratories is researching and developing an application sharing system based on server-based computing that facilitates collaborative work among several participants using unmodified applications. This article gives a brief overview of the technologies.

1. Introduction

A few years ago, many companies, not only information technology (IT) companies, but also non-IT companies, especially manufacturing companies, started to use computers for their tasks. There are many types of collaborative work that is not restricted to one company or one location. For example, car manufacturers make final products using many parts produced by outside suppliers and the building industry often forms joint ventures for large-scale projects.

Collaborative work of this kind requires collaboration platforms constructed over networks interconnecting remote sites to enable remote users to work collaboratively without getting together in one location. Many non-IT organizations have no technical experts who can construct and manage computing systems, so they often outsource the construction and management of their computing systems. Our application sharing system based on server-based computing (SBC) can facilitate collaboration over remote sites interconnected by networks. The management of SBC-based centralized system can easily be outsourced.

2. Server-based computing

SBC facilitates the remote execution of applications by performing almost all computing processes on the application server side (**Fig. 1**). The application is executed on the application server and its windows are sent to the client over a network connection and displayed on the client. The mouse and keyboard events handled by the client are sent to the application server via the network and passed to the application. This data flow allows the application to behave the same as if it were being operated directly on the desktop, but the video cable and keyboard cable have lengthened to become the access network. The CPU, application, and data storage (computing resources) are located on the application servers beyond the access network while the display, keyboard, and mouse (input and output) are on the client. Since very few features are implemented on the client side, the SBC system is also called “a thin client system”.

The SBC system can provide applications designed for desktop PCs at remote sites without needing to modify them. Also almost all the computing resources are located on the application servers and managed centrally, so this can reduce management costs drastically. In addition centralized management can easily enable outsourced services. These days security holes and virus attacks are big problems for client machines such as desktop computers running Microsoft Windows. SBC and thin-client-based systems can reduce the cost of maintaining these clients.

As an alternative to SBC, one common way to provide application service remotely is to use technologies based on the Worldwide Web (WWW). However, this requires the application itself to be WWW-

[†] NTT Information Sharing Platform Laboratories
Musashino-shi, 180-8585 Japan
E-mail: Higuchi.Masafumi@lab.ntt.co.jp

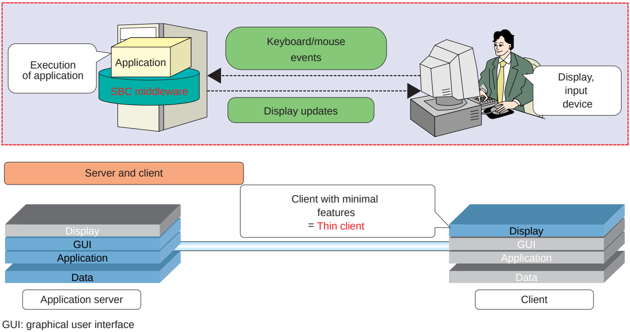


Fig. 1. Overview of SBC.

compatible, so the application often needs to be modified or re-implemented for WWW compatibility. Although new applications can easily be made compatible, it may be very difficult or wasteful to modify existing ones. Furthermore, it is almost impossible to provide graphical applications based on the WWW because of the lack of graphical capability in HTML (hypertext markup language). These difficulties make it impossible to provide some types of application remotely using WWW-based technologies.

3. SBC-based application sharing system

The key features of this system are real-time application sharing, Liberty Alliance single sign on, and session management. These are discussed below.

3.1 Real-time application sharing

In our system, we position a front-end server between the application server where an application program runs and the clients which end-users operate (**Fig. 2**). The front-end server duplicates all the display data that comes from the application server to the clients and distributes the same display views to all the clients simultaneously. It accepts operation information, (e.g., mouse and keyboard events) from all the clients. It chooses one among them and sends the chosen input to the application server. Thus the application program can behave as if it were being operated by a single user.

This system's target applications are graphical applications like computer aided design (CAD). We expect the system to be used where several users are collaboratively editing a single data file. In this case, "several users" means about 2 to 10 people, who are on almost an equal footing. This system facilitates real-time data sharing among remote sites over network connections. Furthermore, since the data is managed as a single instance and several users are sharing and directly manipulating it, there is no need to duplicate or distribute it to the users prior to the collaboration and no need to integrate their results afterwards.

Only one user can operate an unmodified application at one time. Since users' operation are controlled on the system side, users can benefit from collaborative application sharing without the application being modified.

Additionally, to share what part of the display the users are focusing on, this system has an "overlay" feature (**Fig. 3**). The overlay is implemented as a layered transparent window on top of the shared application windows, on which users can draw and type. This additional application supports multiple users simultaneously, so it lets participants communicate with the current user.

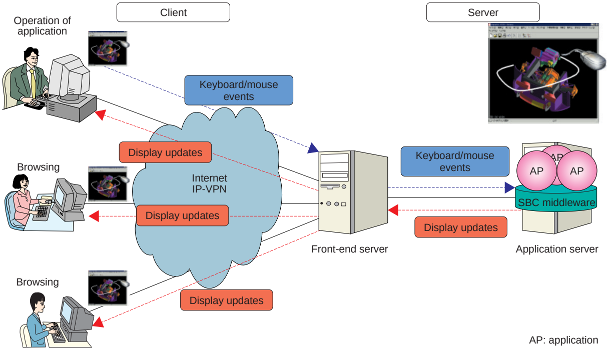
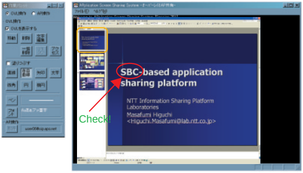


Fig. 2. Architecture of the application sharing system.



SBC-based application sharing platform

NTT Information Sharing Platform Laboratories
Masafumi Higuchi
<Higuchi.Masafumi@lab.ntt.co.jp>

Fig. 3. Overlay feature.

3.2 Liberty-alliance-compliant single sign on

So far, we have described a single application or single application server shared among several users. Our system also lets a single user use several applications installed on one or more application servers. In order to use several applications managed by independent entities, the user may need a different account for each application and must be authenticated for each account when first accessing each application. Our system has a single sign on (SSO) feature that enables users to use all the applications on this system without re-authentication once they have been authenticated by the system (**Fig. 4**). This SSO feature conforms to Liberty Alliance[*1] specifications. Any applications or application servers that comply with Liberty Alliance specifications should be easy to incorporate into our system and use without authentication for each application.

One of the key features of the Liberty Alliance is distributed management of user accounts. Every account is managed in each organization separately, and these accounts are federated on the front-end server. Thus there is no need to create a new account system for SSO: existing account systems can be integrated with this SSO feature.

3.3 Session management

Collaboration using the application sharing features described above is a kind of conference. To support this kind of conference, this system has some session management features.

(1) Session reservation and automatic invitation

Users can reserve a collaborative session and register members who should attend it. Prior to the scheduled time of the session, the system sends an invitation message to each user registered as a session participant.

[*1] The Liberty Alliance is an organization for standardizing an open SSO architecture. Established in September 2001, it currently has over 160 member companies from all over the world. NTT and NTT DoCoMo are among several Japanese companies participating in the Liberty Alliance.

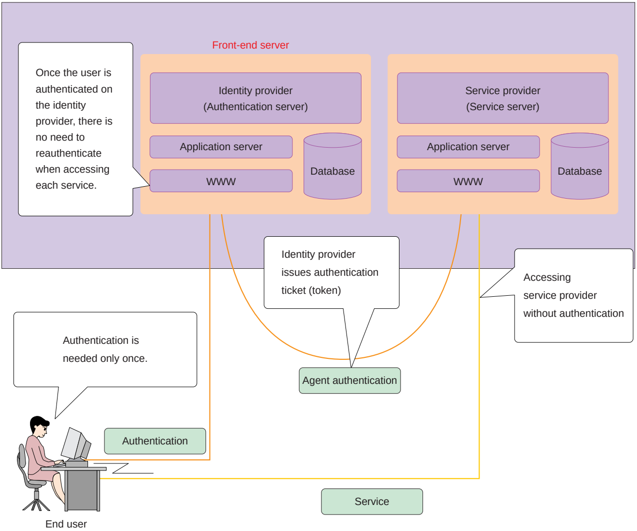

Front-end server
Once the user is authenticated on the identity provider, there is no need to reauthenticate when accessing each service.
Identity provider (Authentication server)
Application server
WWW
Database
Service provider (Service server)
Application server
WWW
Database
Identity provider issues authentication ticket (token)
Accessing service provider without authentication
Authentication is needed only once.
Agent authentication
Authentication
Service
End user

Fig. 4. Single sign on feature.

(2) Session management by SIP

The invitation feature is implemented using SIP (session initiation protocol), which is a signaling protocol that is becoming popular because it is used for VoIP (voice over Internet protocol) applications. It has been standardized in the Internet Engineering Task Force (IETF).

(3) Affinity with other applications

The use of SIP makes this system easy to integrate with other applications, such as voice conferencing and video conferencing, which can be used together with real-time application sharing.

(4) Session directory

The system stores schedule information about the collaboration sessions in a database called the “session directory”. This has user interfaces for registering, reviewing, modifying, and deleting the schedules of collaboration sessions. It also provides a way to view completed or ongoing sessions. Furthermore, the system enable users to participate in an ongoing session directly from the session schedule viewing window.

4. Comparison with related technologies

There are several major SBC products that use Microsoft Windows as application servers: MetaFrame [1] (Citrix), Tarantella [2] (Tarantella), Remote Desktop feature [3] (Microsoft; bundled with Windows XP), and GO-Global [4] (GraphOn). MetaFrame has an application sharing feature. GO-Global plans to provide an application sharing feature with version 3.0. However, none of the SBC products on the market include SSO.

NTT-IT’s application service provider (ASP) network conference service called MeetingPlaza [5], which does not use SBC technologies, features application sharing. However in MeetingPlaza, the application programs are executed on the terminal operated by one of the participants and the other participants share them. In our system, all the application programs are executed only on application servers and the participants share them equally. Moreover, all data is also stored on the application servers. This lets a service provider provide consistent services including application service, service management, and additional services like data storage.

5. Future work

SBC-based systems have all the computing resources centralized on the application server side, so they work well with data center services. They also need full-time connections between servers and clients, so they are technologies of the full-time connection era. SBC technologies are suitable for providing application services to data centers and are useful for managing and operating server systems. They are very important technologies for developing such highly value-added services. We intend to continue working on SBC technologies and to research and develop a wide variety of services.

References

[1] http://www.citrix.com/site/ps/products.asp
[2] http://www.tarantella.com/products/
[3] http://www.microsoft.com/windowsxp/remotedesktop/
[4] http://www.graphon.com/products/GO-GlobalforWindows.shtml
[5] http://www.ntt-it.co.jp/jirei/MeetingPlaza.html

Masafumi Higuchi
Research Engineer, NTT Information Sharing Platform Laboratories.

He received the B.E. and M.E. degrees in instrumentation engineering from Keio University, Yokohama in 1992 and 1994, respectively. He joined NTT in 1994 and researched computer supported cooperative work. He is now working on applications of server-based computing and collaboration software for it.

Manabu Okamoto
Research Engineer, NTT Information Sharing Platform Laboratories.

He received the B.S. and M.S. degrees in mathematics from Waseda University, Tokyo in 1997 and 1999, respectively. In 1999, he joined NTT, Tokyo, Japan and in 2002 he moved to NTT Information Sharing Platform Laboratories, where he is engaged in research on server-based computing and authentication protocols.

Naoaki Shinoda
Engineer, Web Services Project, Application Platform SE Project, NTT Information Sharing Platform Laboratories.

He received the B.S. degree in applied physics from Tokyo Institute of Technology, Tokyo in 1994 and the M.S. degree in physics from the University of Tokyo, Tokyo in 1996. In 1996, he joined the Business Communications Department of NTT's Chiba Branch. In 1999, he joined the Business Communications Headquarters of NTT East Corporation, where he was a member of the team that made the House of Representatives Internet TV Service. He joined NTT Information Sharing Platform Laboratories in 2002. He has received the NTT President's Award and an award from NTT East Business Communications Headquarters.

Naohisa Tanabe
Research Engineer, NTT Information Sharing Platform Laboratories.

He received the B.E. and M.E. degrees in information engineering from Kyushu University, Fukuoka in 1997 and 1999, respectively. In 1999, he joined NTT Information Sharing Platform Laboratories. He has been working on applications of server-based computing and collaboration software for it.