

X ウィンドウシステムにおける疑似サーバ管理システム

島村 政 義^{†,*} 味 岡 義 明[†]

ネットワーク上での分散処理を実現するためにクライアント/サーバモデルを採用したXウィンドウシステムが現在広く普及している。このXウィンドウシステムでは機能拡張の手段として疑似サーバを使用することができる。疑似サーバはXクライアントとXサーバの間に接続され、見かけ上、Xクライアントの機能を拡張する。これによりXクライアントおよびXサーバのいずれも改造することなしに機能拡張を行うことができる。ところがXウィンドウシステム自体は疑似サーバの出現を全く考慮していないため、Xクライアント・疑似サーバ・Xサーバ間の接続をユーザ自身が管理しなくてはならない。例えばユーザは、疑似サーバのサーバ番号を意識して疑似サーバを操作しなければならない。そこで、疑似サーバがより使いやすい環境をユーザに提供することを目的として、この接続関係の管理を自動的に行う「疑似サーバ管理システム」を提案する。疑似サーバ管理システムは、疑似サーバに関する各種の操作を自動化する。本システムによりユーザは疑似サーバのサーバ番号を意識する必要がなくなり、Xウィンドウシステムにおける疑似サーバの利用環境が改善される。

X Pseudo-server Management System in X Window System

MASAYOSHI SHIMAMURA^{†,*} and YOSHIAKI AJIOKA[†]

X Window System is a window system suited for distributed processing because it is based on a client/server model. An X Pseudo-server, which is a process placed between an X client and an X server, can add some functions to the X client without modifying the existing X clients and X servers. However, since the X Pseudo-server is a mere X server for an X client and a mere X client for an X server, the X Window System does not have any particular mechanisms for it. Thus a user by himself must manage any connections between an X client, X Pseudo-server and X server. For example, he must use a different server number for the X Pseudo-server from the X server if they work on the same computer. Therefore, we propose X Pseudo-server Management System which automatically connect between an X client, X Pseudo-server and X server. This system makes us use X Pseudo-server very easily.

1. はじめに

一般にワークステーションは次の2つの特徴を持つ。すなわち、ビットマップディスプレイを持つこと、およびコンピュータネットワークに対応していることである。このようなワークステーションの特徴を生かしたウィンドウシステムとして、MITが開発したXウィンドウシステム¹⁾がある。従来のウィンドウシステムにはネットワークによる分散処理を考慮していないものが多かったが、Xウィンドウシステムはクライアント/サーバモデルを採用することにより分散処理に対応している。今日ではXウィンドウシステム

は業界標準の地位を獲得している。

Xウィンドウシステムでは、アプリケーションを実際に行うXクライアントとキーボードやディスプレイなどを管理するXサーバが分離し、両者がネットワークを介してXプロトコル²⁾に従い通信する。したがって既存のアプリケーションに新たな機能を追加するためには、Xクライアントを改造するか新たなXクライアントを作る必要がある。しかしXサーバやネットワークに関する機能を拡張するのならば、疑似サーバ³⁾(XPS: X Pseudo-server)を用いることもできる。

疑似サーバとはXクライアントとXサーバの間に接続されるプログラムであり、見かけ上、Xクライアントの機能を拡張する。疑似サーバは、Xクライアントからは単なるXサーバに見え、Xサーバからは単なるXクライアントに見える。疑似サーバを用いることにより、XクライアントおよびXサーバのいずれも改造

[†] 豊田工業大学工学部
Department of Engineering, Toyota Technological Institute

* 現在、富士通株式会社企画本部クライアントサーバシステム企画統括部
Presently with CSS Strategy and Alliance Div., IT Strategy and Planning Group, Fujitsu Limited

することなしに機能拡張を行うことができる。

しかし、Xウィンドウシステム自体には疑似サーバを考慮した機構は組み込まれていない。このため、Xクライアント・疑似サーバ・Xサーバ間の接続をユーザ自身が管理しなくてはならない。例えば、すべての疑似サーバにはサーバ番号と呼ばれる識別番号を与える必要があるが、この作業はユーザが行わなければならない。また、サーバ番号が同一マシン内で重複しないようにユーザが管理する必要がある。このように疑似サーバはXクライアントの機能を拡張する上で便利ではあるが、操作性が悪く使いにくい面もある。

このような問題を解決するため、疑似サーバ管理システムを提案する。その実現例として、ウィンドウマイグレーション機能を実現する疑似サーバ Inter-X⁴⁾を疑似サーバ管理システムと組み合わせ、疑似サーバの使いやすい環境を実現する。従来はXクライアント・疑似サーバ・Xサーバ間の接続方法が疑似サーバごとにまちまちであったが、本システムはこの接続方法を統一し、かつ自動化する。したがって疑似サーバに対するサーバ番号の割り当ても自動的に行われるようになる。また、ユーザによる疑似サーバの操作方法も従来はまちまちであったが、本システムではこれも統一する。本論文では、この疑似サーバ管理システムの概要を述べ、その有用性を確認する。

2. 疑似サーバとその問題点

Xウィンドウシステムの機能を拡張する手段の1つとして疑似サーバがある。疑似サーバはXクライアントとXサーバの間に入ることにより、例えば以下のような機能を実現する。

- Xプロトコルの解析と表示 (図1(a): xscope⁵⁾)
XクライアントとXサーバ間の通信に用いられるXプロトコルを解析し表示する。
- ウィンドウの共有 (図1(b): xmx⁶⁾, xtv⁷⁾)

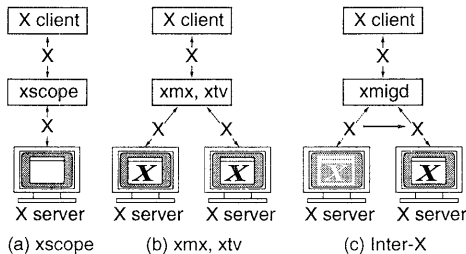


図1 各種の疑似サーバ

Fig. 1 Examples of X Pseudo-servers.

ある1つのウィンドウを複数のXサーバにコピーして表示し、複数のユーザで共有可能にする。

- ウィンドウマイグレーション (図1(c): Inter-X)
あるXサーバから別のXサーバにウィンドウを移動させる。

疑似サーバはXクライアントやXサーバに全く手を加えずにこれらの機能を実現することができるという特徴を持つ。疑似サーバの作成者のために、疑似サーバを作るためのライブラリ⁸⁾も用意され始めている。

しかし、Xウィンドウシステムには疑似サーバを使用するための特別な機能は用意されていない。つまり疑似サーバの存在自体が考慮されていないのである。このため次のような問題が発生している。

- Xクライアント・疑似サーバ・Xサーバ間の接続手順が統一されていない。
 - ユーザがXクライアント・疑似サーバ・Xサーバ間の接続関係を把握することが難しい。
 - 疑似サーバが使うサーバ番号が重複する。
 - 疑似サーバに対する操作方法が統一されていない。
- このような問題は疑似サーバの使い勝手を悪くし、疑似サーバの開発者とユーザに余分な負担を掛けている。

3. 疑似サーバ管理システムの提案

3.1 必要とされる機能

このような問題を解決するために、疑似サーバ管理システムを提案する。疑似サーバを使いにくくしている要因を取り除くために、疑似サーバ管理システムは次のような機能を備えなくてはならない。

- Xクライアント・疑似サーバ・Xサーバ間の自動接続
- ユーザに対する、Xクライアント・疑似サーバ・Xサーバ間の接続関係の報告
- 疑似サーバが使うサーバ番号の自動管理
- 疑似サーバに対する操作方法の統一

疑似サーバ管理システムは、これらの機能を使いやすい形でユーザに提供しなければならない。また、疑似サーバ管理システムの運用がシステム管理者にとって大きな負担となってはならない。これらの点に留意して疑似サーバ管理システムの設計を行った。

3.2 システム構成

本システムは次の3つのプログラム、すなわち疑似サーバ・コントロールパネル・XPSM (X Pseudo-server Manager) から構成される。以下にそれぞれ

のプログラムについて説明する。

3.2.1 疑似サーバ

疑似サーバは、その疑似サーバ独自の機能のほかに、疑似サーバ管理システムの XPSM と通信する機能を持つものとする。これは後述する XPSMP (XPSM Protocol) により可能となる。また、本システムは統一された操作環境を提供する必要がある。そのため、個々の疑似サーバごとにユーザインタフェースを持つのではなく、次に説明するコントロールパネルと呼ばれるプログラムがユーザインタフェースを担当する。

3.2.2 コントロールパネル

すべての疑似サーバに対して統一した操作環境を実現するために、ユーザはコントロールパネルと呼ばれるプログラムを通してあらゆる操作を行う。ユーザは個々の疑似サーバに対する操作にも、本システムに対する操作にも、必ずコントロールパネルを用いる。

ユーザはコントロールパネルを用いることにより、Xクライアントと疑似サーバの立ち上げや、疑似サーバに対する遠隔操作、接続状況の把握などができる。コントロールパネルはネットワーク内のどのコンピュータ上で動作してもよく、1ユーザが同時に複数のコントロールパネルを動作させてもよい。コントロールパネル自体の起動や終了はXクライアントや疑似サーバには影響を与えない。

コントロールパネルにこのような柔軟性を与えるために、Xクライアント・疑似サーバ・Xサーバ間の接続に関する情報は次に説明する XPSM と呼ばれるプログラムが管理する。つまり本システムでは、コントロールパネルはユーザからの指示を仲立ちして XPSM に送るクライアントであり、XPSM はユーザからの指示を実際に行うサーバである。

3.2.3 XPSM

Xクライアント・疑似サーバ・Xサーバ間の接続に関する情報をコントロールパネル自身が管理すると、コントロールパネルが重いプログラムになってしまうばかりか、ネットワーク上のどのコンピュータからでも指示を出せるといった柔軟性を実現することが難しくなってしまう。そこで接続情報は XPSM と呼ばれるプログラムが管理することにする。

XPSM は接続情報を管理すると同時にコントロールパネルと疑似サーバとの間の仲立ちを行う。すなわち、コントロールパネルからの指示に従って、Xクライアントと疑似サーバの立ち上げや、疑似サーバへの

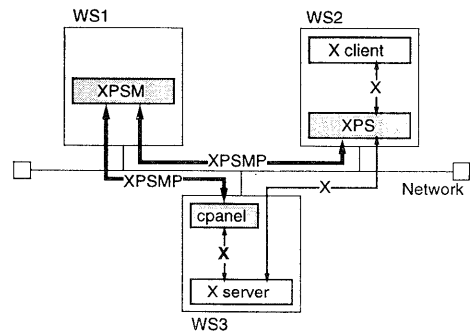


図 2 疑似サーバ管理システムの構成

Fig. 2 Structure of X Pseudo-server Management System.

コマンドの発行を行う。XPSM プロセスは予めネットワーク内の任意のコンピュータ上に1つ動作させておく。コントロールパネルが XPSM にアクセスすると新たにそのユーザ専用の XPSM が自動的に生成される。

3.2.4 各プログラムの接続

疑似サーバ・コントロールパネル・XPSM および Xクライアント・Xサーバは図2に示すように接続する。各プロセスは同一のコンピュータ上で動作してもよいし、すべて別々のコンピュータ上で動作してもよい。これらのプログラム間の通信には、Xプロトコルと、後述する XPSMP を用いる。

Xプロトコルを用いるのは次のプログラム間の通信である。

- Xクライアント⇔疑似サーバ
- 疑似サーバ⇔Xサーバ
- コントロールパネル⇔Xサーバ

XPSMP を用いるのは次のプログラム間の通信である。

- コントロールパネル⇔XPSM
- XPSM⇔疑似サーバ

3.3 XPSMP

本システムを構成するプログラムのうち、コントロールパネルと XPSM および XPSM と疑似サーバは、ユーザからの指示などを伝えるために通信を行う必要がある。本システムでは、この通信に XPSMP と呼ばれる独自のプロトコルを用いる。このプロトコルはXプロトコルを参考にして設計した。このプロトコルはXプロトコルと同様に8ビット幅の非同期な全二重式バイトストリームを用いることを前提としており、今回はバークレイソケット⁹⁾を用いて実現した。

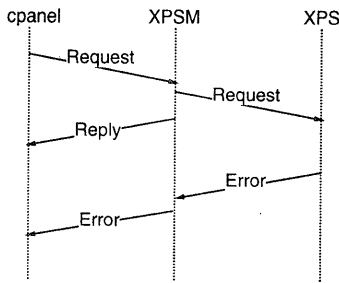


図 3 XPSMP のプロトコルシーケンス

Fig. 3 Protocol sequence of XPSMP (X Pseudo-server Manager Protocol).

表 1 XPSMP のプロトコルフォーマットの一例
Table 1 An example of a protocol format of XPSMP (X Pseudo-server Manager Protocol).

バイト数	型	内 容
1	unsinged int	code (06 d)
3		non use
64	char	user name
4	unsinged int	user ID
4	unsinged int	operation code
64	char	X server machine
4	unsinged int	server no.
4	unsinged int	screen no.

3.3.1 プロトコルシーケンス

XPSMP で規定しているメッセージはリクエスト、リプライ、エラーの3種類である(図3参照)。コントロールパネルから XPSM へ、あるいは XPSM から疑似サーバへ送られるメッセージがリクエストであり、XPSM からコントロールパネルへの応答がリプライである。また、疑似サーバから XPSM へ送られ、さらにコントロールパネルへと送られるのがエラーである。ユーザからの指示はリクエストとなり、コントロールパネルから XPSM へ、XPSM から疑似サーバへと伝わる。

3.3.2 プロトコルフォーマット

XPSMP では 13 種類のリクエストと 7 種類のリプライ、そして 2 種類のエラーのフォーマットを規定している。特定のアーキテクチャに依存しないために、フォーマット中のデータは 32bit 境界に従って並べられている。XPSMP のフォーマットの一例を表1に示す。

3.3.3 立ち上げシーケンス

疑似サーバと X クライアントの立ち上げは次のような手順で行う。

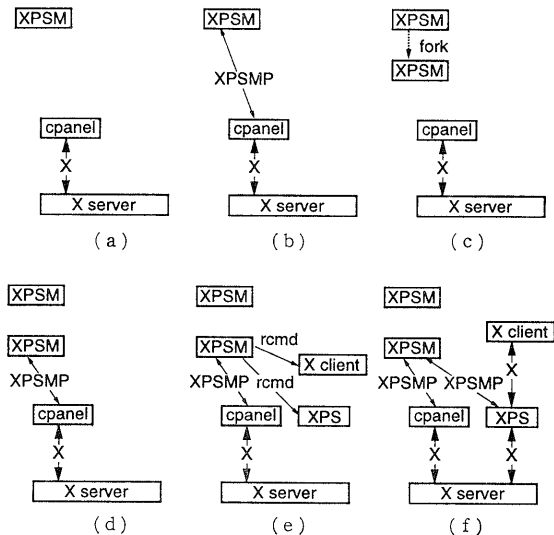


図 4 Xクライアントと疑似サーバの立ち上げシーケンス

(a)コントロールパネルの起動, (b)子 XPSM のポート番号の問い合わせ, (c)子 XPSM の生成, (d)コントロールパネルと子 XPSM との接続の確立, (e)コントロールパネルによる立ち上げ指示, (f)Xクライアントと疑似サーバの接続

Fig. 4 Sequence of starting X client and X Pseudo-server.

(a)A user starts a control panel. (b)The control panel inquires a port number of a child XPSM from a parent XPSM. (c)The parent XPSM creates the child XPSM with a fork function. (d)The control panel is connected with the child XPSM. (e)A user orders that an X client and XPS are created by the child XPSM with the control panel. (f)The X client is connected with the XPS.

a)コントロールパネルの起動

システム管理者は予め親 XPSM を起動しておく。これはネットワーク内の任意のコンピュータ上に1プロセスだけでよい。一方、ユーザは、コントロールパネルをネットワーク内の任意のコンピュータ上で起動する(図4(a))。

b)子 XPSM のポート番号の問い合わせ

コントロールパネルは親 XPSM に対してそのユーザ専用の子 XPSM のポート番号を問い合わせる(図4(b))。

c)子 XPSM の生成

そのユーザ専用の子 XPSM がまだなければ、親 XPSM が fork して生成する(図4(c))。

d)コントロールパネルと子 XPSM との接続の確立

コントロールパネルは子 XPSM と接続する(図4

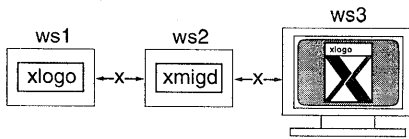


図 5 各プログラムの接続の例

Fig. 5 An example of connected an X client, X Pseudo-server and X server.

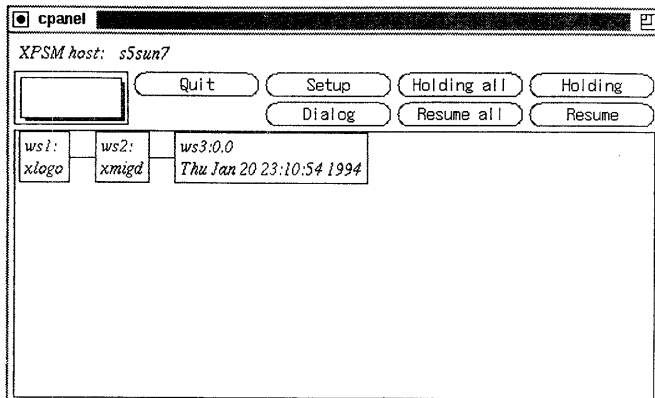


図 6 接続状況の表示

Fig. 6 Indication of connection status.

(d).

e) コントロールパネルによる立ち上げ指示

ユーザはコントロールパネルを用いてXクライアントと疑似サーバの立ち上げを指示する。この指示に従って子 XPSM はリモートコマンド⁹⁾ (rcmd: remote command system call) を用いて立ち上げを行う。(図 4 (e)).

f) Xクライアントと疑似サーバの接続

まず疑似サーバがXサーバと接続し、それからXクライアントが疑似サーバと接続する。疑似サーバが用いるサーバ番号の処理などは子 XPSM が自動的にを行う(図 4 (f)).

3.4 実現された機能

本システムの有用性を確認するため、本大学の一柳が作成した疑似サーバ Inter-X (プログラム名は xmigd) を本システムに対応させた。これを例にして、以下に実現された機能の一部を述べる。

3.4.1 接続状況の把握

本システムでは、コントロールパネル (cpanel: control panel) と呼ばれるプログラムがユーザとのインタフェースを担当する。このコントロールパネルはXクライアント・疑似サーバ・Xサーバ間の接続関係

をユーザに報告する機能を持つ。例えば図 5 に示すように、ワークステーション 1 にXクライアント “xlogo” が、ワークステーション 2 に Inter-X の疑似サーバ “xmigd” が動作しており、xlogo のウィンドウがワークステーション 3 のXサーバに表示されたい。この場合、コントロールパネルには図 6 のように表示される。ユーザはこの表示により各プログラムの接続関係を把握することができる。

3.4.2 疑似サーバとXクライアントの立ち上げ

従来の疑似サーバの使用方法では、図 5 に示したように接続するためにコマンドラインから次のような入力を与える必要がある。まず xmigd が用いるサーバ番号を各自が適当に決めなければならない。ここではサーバ番号を 1 番とし、xmigd のオプションに -p 1 を指定する。またウィンドウの出力先は -display ws3: 0 と指定する。-display の引数の形式は次のとおり。

host : *server*

host はXサーバ (または疑似サーバ) が動作しているマシンの名前であり、*server* はXサーバ (または疑似サーバ) のサーバ番号である。したがって次のようにキー入力する。

```
rsh ws2 xmigd -p 1 -display ws3: 0
```

なおスクリーン番号は説明を簡単にするために省略している。次に xlogo の接続先サーバ番号を xmigd に合わせて 1 番とするため、オプションに -display ws2: 1 を指定する。したがって次のようにキー入力する。

```
rsh ws1 xlogo -display ws2: 1
```

一方、疑似サーバ管理システムでは、コントロールパネルが表示する図 7 のようなダイアログボックスを使う。ほとんどの入力項目にはデフォルト値が設定されるため、ユーザはマウスによって起動したいXクライアントと疑似サーバを選択するだけでよい。Xクライアント・疑似サーバ・Xサーバ間の接続やサーバ番号の管理は自動的にされるため、ユーザがサーバ番号を決める必要はない。本システムに対応したすべての疑似サーバは、このダイアログボックスにより立ち上げを行うことができる。

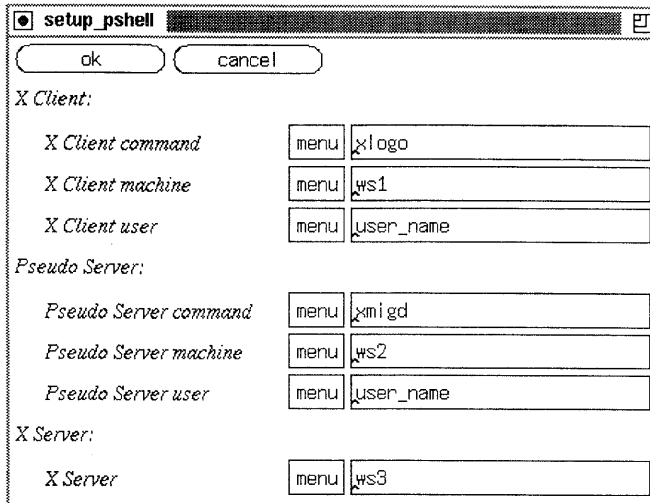


図 7 ダイアログボックス
Fig. 7 Dialogbox.

4. 考 察

4.1 必要とされる資源

疑似サーバ管理システムが必要とする計算機資源は比較的小さい。表 2 に示すような環境で測定した所要のメモリ容量を表 3 に示す。扱う疑似サーバの個数が増えても、メモリ容量が急激に増加するようなことはない。

表 2 消費資源測定環境

Table 2 The environment when the resources is measured.

コンピュータ	Sun SPARC station ELC
OS	SunOS Release 4.1.1
コンパイラ	cc (最適化オプション付)
リンク方式	ダイナミックリンク
Xウィンドウシステム	X11R5

表 3 疑似サーバ管理システムが必要とするメモリ容量
Table 3 Memory requirement the X Pseudo-server Management System.

疑似サーバの個数	XPSM [Kbyte]	cpanel [Kbyte]
0	28	332
1	36	344
2	36	348
3	40	348
5	40	356
10	44	368
15	48	384

コントロールパネルの起動や終了は XPSM や疑似サーバの動作に影響しない。このため、ユーザが必要とする時以外はコントロールパネルを終了させておくことで計算機資源を節約することもできる。

また、疑似サーバ管理システムが必要とする CPU の使用量も小さい。というのも、ユーザがコントロールパネルを用いて指示を出すなどのイベントがあった時以外は XPSM もコントロールパネルも入力待ちをしているだけであり、CPU をほとんど使わないからである。

4.2 Xクライアントの描画性能に与える影響

本システムや疑似サーバは Xクライアントの描画性能に対して次のような影響

を与える。

a) 疑似サーバ管理システムによる影響

疑似サーバが XPSM と通信を行うのは、主にユーザがコントロールパネルを用いて疑似サーバに指示する時だけである。このため本システムを用いることにより疑似サーバや Xクライアントが本来持っている処理能力が低下するようなことはない。

b) 疑似サーバによる影響

本大学の一柳は、疑似サーバが Xクライアントの描画性能にどのような影響を与えるかを調べるために、疑似サーバを経由して接続した Xサーバが単位時間当たり描画できる要素数を測定した¹⁰⁾。疑似サーバには Inter-X を、マシンには Fujitsu S-4/IP を用いた。測定結果を図 8 に示す。図の縦軸は描画要素を示し、横軸は Inter-X を経由しない場合の描画性能に対する Inter-X を経由した場合の描画性能比である。図中のローカル接続 (local connection) とは Xクライアントと Inter-X が同一のマシン上で動作している場合であり、リモート接続 (remote connection) とは Xクライアントと Inter-X が別々のマシン上で動作している場合である。リモート接続の場合、Xクライアントの描画性能はあまり低下しない。ローカル接続の場合でも、描画性能の低下を 50% 以内に抑えることができる。

4.3 本システムの適用範囲

本システムは接続先の Xサーバが必要に応じて切り替わるような疑似サーバ (例えば Inter-X) や、一度

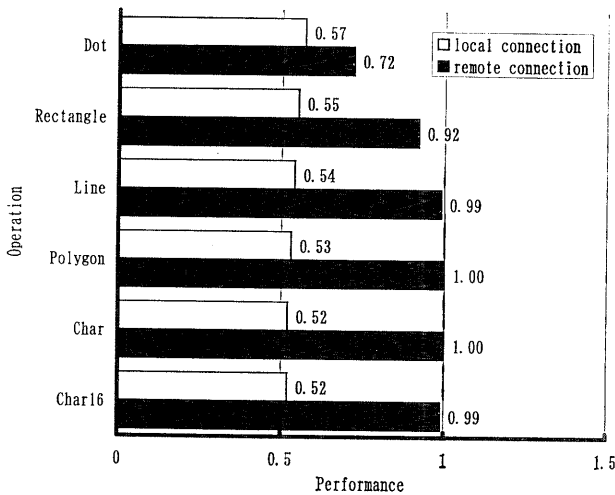


図 8 Inter-X を経由した際の描画性能比
Fig. 8 Relative graphics performance of Inter-X.

に数台の Xサーバが接続されるような疑似サーバを管理することができる。しかし、疑似サーバがさらに別の疑似サーバに接続するような複雑な接続を行う疑似サーバを管理することはできない。

4.4 本システムの持つ意味

疑似サーバを用いることでウィンドウのマイグレーションやウィンドウの共有が可能になる。図 9 (a)は疑似サーバを用いて異なる Xサーバ間で同一のウィンドウを共有する例である。しかし疑似サーバを接続することにより通信効率が悪化する。これを避けるため、特に有用な疑似サーバの機能は Xクライアント側で実現する場合もある。図 9 (b)はこれまで疑似サー

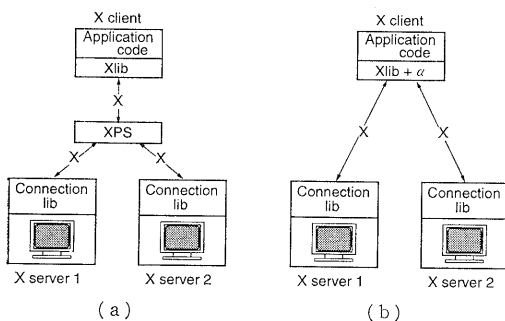


図 9 ウィンドウ共有の実現例
(a)疑似サーバによるウィンドウ共有, (b)Xクライアントの拡張によるウィンドウ共有
Fig. 9 Examples of window sharing.
(a)A window sharing by an XPS. (b)A window sharing by an extended X client.

バで実現していた機能を Xクライアント側のライブラリの拡張で実現した例である。

しかしながら、このように Xクライアント側のライブラリを拡張すると通信効率の悪化は免れるもののライブラリの肥大化をもたらす、Xクライアントのオブジェクトコードサイズを増大させる。事実、Xクライアントのオブジェクトコードサイズの拡大は問題になっており¹¹⁾、数 Mbyte の Xクライアントも珍しくなくなっている。このような方法で新たなネットワーク機能を追加していくのは本質的な改善方法ではないように思われる。

もともと Xクライアントはただ 1つの Xサーバとだけ接続するのが Xウィンドウシステムの基本的な考え方である。この基本を守りながら Xウィンドウシステムのネットワーク機能を拡張するための方法として図 10 (a)のように Xクライアントおよび Xサーバからネットワーク機能を分離する方法が考えられる。

Xクライアントからは Xlib を拡張・分離し、Xサーバからは通信ライブラリ (図中では Connection lib) を拡張・分離する。これまで疑似サーバなどで実現されていたウィンドウの共有・Xクライアントの保留・Xクライアントの再開などの機能はこの拡張 Xlib・拡張通信ライブラリが行う。

図 10 (b)に示すように Xクライアントや Xサーバ

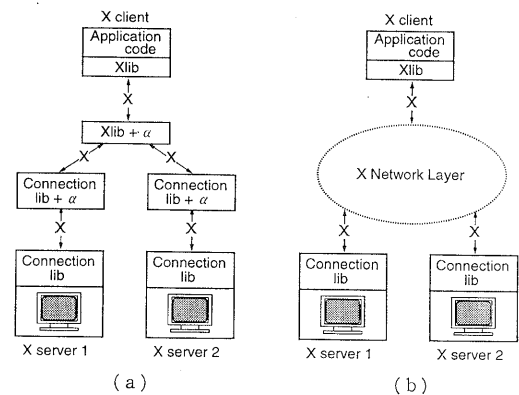


図 10 Xネットワーク層の提案
(a)通信ライブラリの分離, (b)Xネットワーク層
Fig. 10 Proposal of X Network Layer.
(a)A separated communication function from an X Server and X Client. (b)The separated communication function makes an X Network Layer.

からは拡張 Xlib・拡張通信ライブラリ層は単一のネットワーク層（ここではXネットワーク層と呼ぶことにする）と見ることができる。XクライアントやXサーバはこのXネットワーク層に接続しさえすればよく、接続状態の詳細を気にする必要はなくなる。

このようなXネットワーク層ができた場合、Xネットワーク層の接続管理システムが必要となる。本研究における疑似サーバ管理システムが発展したものが、このXネットワーク層における接続管理システムとなるであろう。現在の疑似サーバ管理システムはこのXネットワーク層における接続管理システムの前段階にあるものと考えることができる。

5. おわりに

疑似サーバ管理システムを用いることにより、疑似サーバの利用環境が改善されることを示した。今後の課題としては、本システムとウィンドウマネージャ(WM: Window Manager)の融合が挙げられる。というのは、ユーザがウィンドウを操作するためのプログラムとしてXウィンドウシステムには既にウィンドウマネージャがある。これとは独立して本システムを付け加えた結果、図11(a)に示すようにウィンドウを操作するためのプログラムがコントロールパネルとウィンドウマネージャの2つになってしまった。ユーザは疑似サーバ関連の操作にはコントロールパネルを、それ以外のウィンドウ操作にはウィンドウマネージャを用いなければならない。この使い分けはユーザにとって煩わしい。本来ならば図11(b)に示すようにウィンドウマネージャと融合すべきであろう。

また、本システムで疑似サーバを管理するためには、疑似サーバ側が本システムに対応していなければ

ならない。この対応を容易にするために、疑似サーバ作成者向けのライブラリやドキュメントを用意した。現在のところ本システムに対応している疑似サーバは本学の一柳が開発したInter-Xのみであるので、今後はライブラリをより充実させ、多くの疑似サーバを本システムに対応させていかなければならないだろう。

謝辞 本研究を進めるにあたり、貴重な助言を頂いた豊田工業大学情報システム研究室の北川一教授、中川徹助教授、一柳勝義氏に感謝します。

参考文献

- 1) Scheifler, R. W. and Gettys, J.: The X Window System, *ACM Trans. Graphics*, Vol. 5, No. 2, pp. 79-109 (1986).
- 2) Nye, A.: *X Protocol Reference Manual*, O'Reilly & Associates, Inc. (1988) (邦訳: Xプロトコル・リファレンス・マニュアル, 石川和也訳, ソフトバンク (1991)).
- 3) 渡部 勇, 神田陽治: ユーザインタフェース大作戦4, *bit*, Vol. 24, No. 4, pp. 410-419, 共立出版 (1992).
- 4) 一柳勝義, 北川 一: Xウィンドウ・マイグレーション機構とその一実現例, 第47回情報処理学会全国大会論文集, No. 5, pp. 145-146 (1993).
- 5) Peterson, J.F.: *xscope—X Window System Protocol Monitor*, X 11 R 5 contrib distribution (1991).
- 6) Bazik, J.: *XXM: An X Protocol Multiplexor, Version 1.0*, Brown University (1990).
- 7) Abdel-Wahab, H.M. and Feit, M.A.: *A Framework for Sharing X Window Clients in Remote Synchronous Collaboration*, Old Dominion University (1991).
- 8) Menges, J.: *The X Engine Library: A C++ Library for Constructing X Pseudo-server*, THE X RESOURCE, Issue Five, pp. 129-141, O'Reilly & Associates, Inc. (1993).
- 9) Stevens, W.R.: *UNIX NETWORK PROGRAMMING*, Prentice Hall, Inc. (1990) (邦訳: UNIX ネットワークプログラミング, 篠田陽一訳, トップラン (1992)).
- 10) 一柳勝義: Xウィンドウマイグレーション機構の方式とその実現に関する研究, 平成5年度修士論文発表会要旨集, pp. 5-1-5-2, 豊田工業大学 (1994).
- 11) 太田昌孝: Xウィンドウはいつ亡びるか, *bit* 別冊 Xウィンドウとその仲間たち, pp. 201-205, 共立出版 (1992).

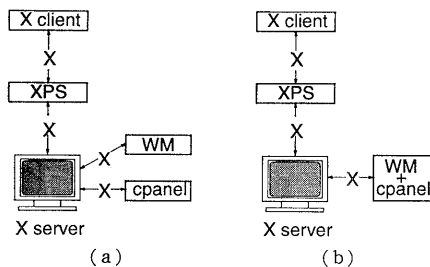


図11 コントロールパネルとウィンドウマネージャ
(a)現状のコントロールパネル, (b)ウィンドウマネージャと融合したコントロールパネル

Fig. 11 The control panel and window manager.
(a) An actual situation of a control panel.
(b) A control panel integrated into a window manager.

(平成6年3月29日受付)

(平成6年9月6日採録)

**島村 政義 (正会員)**

1967年生。1987年桐蔭学園工業高等専門学校工学部電気工学科卒業。同年富士通株式会社に入社。1990年豊田工業大学に入学。1994年同大学工学部制御情報工学科卒業。現在、富士通企画本部クライアントサーバシステム企画統括部に所属。X/Open や OMG への提案の立案、動向調査、プロトタイプを試作等に従事。OMG に継続的に出席中。

**味岡 義明 (正会員)**

1965年生。1987年慶應義塾大学理工学部電気工学科卒業。1992年同大学院理工学研究科博士課程修了。博士(工学)。現在、豊田工業大学工学部制御情報工学科講師。ニューラルネットワークを中心とした数理モデルを軸に脳の高次情報処理の解明を試みている。またこれにともなって人間の情報処理の仕方にも興味を持ち、ユーザインタフェースやマン・マシンコミュニケーション、さらにはコンピュータネットワークの研究にも従事している。IEEE, INNS, 日本神経回路学会, 電子情報通信学会各会員。