2P−5

位置透過性を提供する統一されたユーザ環境の実現

小口 和弘*　多田 好克†
電気通信大学 大学院 情報システム学研究科‡

1　はじめに

コンピュータの普及によって、ユーザがそれを使える場所は増加している。しかし、それぞれの場所ごとにユーザ環境は異なっている。これは、ある場所で使えるアプリケーションが別の場所では使えないという問題を引き起こす。そこで本研究では、ユーザが移動しても同じユーザ環境を提供するシステムを実現する。本システムは移動先からより近い場所にある資源を使って移動時の環境を再構築する。近い資源を使うことで、遠隔地ログインをして離れた場所にあるユーザ環境を使う場合よりもアプリケーション使用時の応答性が向上する。

2　位置透過なユーザ環境

このシステムの目的は、移動の前後で同じアプリケーションが使え、所有しているファイルが見えるという位置透過なユーザ環境を実現することである。

このような環境の実現方法としては、ユーザが常に使用しているコンピュータに遠隔地ログインをするか、携帯型コンピュータに環境を構築してそれを持ち歩くことが考えられる。しかし、遠く離れたコンピュータに遠隔地ログインした場合、遅延が大きく、アプリケーションの応答 (例えばエコーバックの時間) が悪くなり、使い難くなってしまう。携帯型コンピュータを使う場合、携帯サイズからくるハードウェアの制約により、十分な環境を構築できないという問題がある。さらに、コンピュータを持ち歩かなければならないのでユーザに負担をかけることにもなる。そこで、本システムでは、アプリケーションの応答が良く、移動の際ユーザに負担がかからない環境の実現も目的とする。

An Implementation of Unified User Computing Environment Provided with Location Transparency.
*Kazuhiro Oguchi
†Yoshikatsu Tada
‡Graduate School of Information Systems, The Univ. of Electoro-Communications.

3　システムの概観

このシステムは、ユーザが移動先に既設のコンピュータを使い、その場に自分の環境を構築する。アプリケーションは、移動先のコンピュータで移動前と同じ物が検索され、これがその場で実行される。そこで、アプリケーションが見つからない場合、より近いコンピュータ上にあるものが検索され、実行される。移動先から近い場所のアプリケーションが使われることで、応答性が向上する。ユーザの所有するファイルも、移動先からより近いコンピュータ上にキャッシュすることで、ファイルI/Oの向上を図る。

システムは図1に示すように、クライアント、実行サーバ群、ホームサーバから成る。

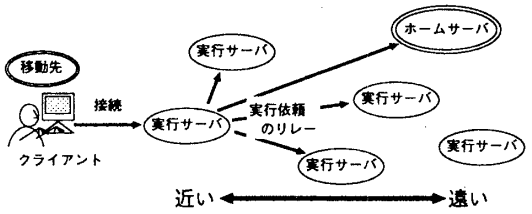

図1：システムの概観

クライアント

クライアントは一番近い実行サーバを検索して、これに接続する。実行されるアプリケーションは、まずそのサーバ上で検索・実行される。アプリケーションはクライアントの端末を通して入出力をおこなう。クライアントはJavaApplet[1]で実装した。これにより、ユーザはWWWブラウザが使える場所なら、どこからでもシステムを使えるようになる。

実行サーバ群

実行サーバは複数あり、あるサーバで実行できないアプリケーションは、別のサーバに実行依頼がリレーされる。アプリケーション実行依頼のリレーはクライアントに近いサーバから順におこなわれる。近いサーバの基準は4節で述べる。

ホームサーバ

ホームサーバはユーザが使っているアプリケーションのリストとユーザの所有するファイルを管理する。アプリケーションのファイルI/Oは、最終的にホームサーバに対しておこなわれる。また、ホームサーバは実行サーバでもあるので、クライアントから近い実行サーバに無いアプリケーションは、最終的にはホームサーバで実行される。よって、ユーザが使うアプリケーションはホームサーバで必ず実行できるものでなければならない。

4 実行サーバの探索・決定

本システムを実現する際に、次の問題があった。

- Appletのネットワーク接続が困難
- 予め分かっていない実行サーバの検索・決定
- アプリケーションの同一保証が移動前後で困難
- アプリケーションのファイルI/Oの効率が悪い

本稿では、実行サーバの検索と決定について述べる。

アプリケーションの実行はクライアントから近い実行サーバ順におこなわれるので、クライアントは近い順に並んだ実行サーバのリストを持っている。

以下、このリストの作成手順について述べる。

4.1 実行サーバの探索

このシステムでは、実行サーバは、広範囲にあまねく存在していて、使えるサーバの数やアドレスは事前には分からないものとしている。そこで、オンデマンドに少ない通信量でサーバを探すために、IP Multicast[2]を使うことにした。

Multicastでは、IPヘッダ中のTTL(TIME TO LIVE)値を設定してMulticastグループに対してパケットを送出する。TTLはパケットを中継するMulticastルータを通る度に減算されていき、値が0になると、そのパケットは捨てられる。つまり、設定したTTLの値が送出元からのパケット最大到達範囲の距離となる。

そこで、TTLをいくつに設定したかという情報をパケットに記録して、TTLを1から順に増やしながら送出する。すると、それを受け取ったホストまでの距離は、最初にそのホストに到達したパケットに記録しておいたTTLの設定値をみればわかる。

この仕組みを用いて、クライアントとホームサーバまでの距離の値(距離C-Hと呼ぶ)と、クライアントを中心とした、距離C-Hの範囲内の実行サーバとの距離がわかる。

4.2 実行サーバの決定

決定される実行サーバは、ホームサーバからも近い方が良い。これは、実行サーバがファイルI/Oをするためである。

そこで、今度はTTLの値を距離C-Hに設定して、ホームサーバからパケットを送出する。すると、ホームサーバを中心とした距離C-Hの範囲内の実行サーバがわかる。

そして、クライアントとホームサーバそれぞれからの実行サーバ探索結果の積集合を使用する実行サーバとする。これは図2に示す範囲にある実行サーバである。

この範囲の実行サーバは、クライアントまたは、ホームサーバに対して距離C-H内にある。

クライアントはこの範囲の実行サーバをクライアントから近い順に使っていく。

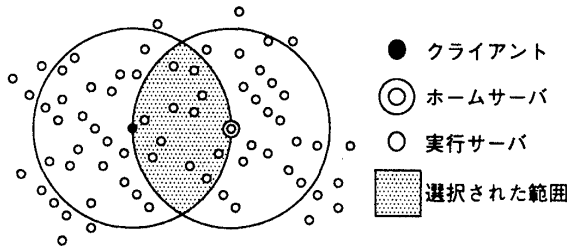

図2：実行サーバの探索

5 まとめ

移動前後で同じユーザ環境を使えるようにするために、本研究では、移動先で同じ環境を動的に再構築する手法を取った。さらに、移動先から近い場所にあるサーバ上の資源を使うことで応答性を向上させた。また、近いサーバを探索・決定する手法を示した。

参考文献

[1] JDK1.1 Documentation, http://java.sun.com/products/jdk/1.1/docs/.

[2] W.Richard Stevens,TCP/IPIllustrated vol.1, Addison-Wesley,1994.