

X-window 上の Smalltalk

小方 一郎
電子技術総合研究所

ネットワーク上の資源の共有という立場から、ビットマップやマウスに関してネットワーク透明な X-window システムが注目を浴びている。また、Smalltalk-80 は汎用機の上で使えるようになったとはいえ、なお一層の高速実行を求められている。専用ハードウェアの開発が盛んなのもこのためである。来るべき専用ハードウェアをネットワーク資源として共有する可能性を調べるため、X-プロトコルによる、Smalltalk-80 インタプリタの実装を試みてみた。まだこのインプリメンテーションは十分な性能を得てはいないが、一応の経験が得られたのでここに報告する。

Smalltalk-80 interpreter on X-window

Ichiro OGATA
Electrotechnical Laboratory
1-1-4 Umezono, Sakura, Niihari, 305 JAPAN

Recently, Smalltalk-80 can be executed on general purpose machine. On the other hand, it is desired to execute Smalltalk-80 much faster. For that purpose, some special hardware for Smalltalk-80 has been developed.

To share the such kind of special CPU resources, X-window gives us one solution. So, I installed a new smalltalk interpreter on X-window system to research the feasibility such resource sharing.

1. まえがき

Smalltalk-80 は、個人用の専用計算機のういで用いられるべく設計された。しかし、最近の研究により汎用ワークステーションの上でも十分な速度での実行が可能となっている。筆者が開発に携わった 菊32V もこの流れの中のインプリメンテーションであった。最近のワークステーションの速度向上の結果、菊32V でも xerox の Smalltalk 専用マシン Dorado とほぼ同じ程度の高速実行が可能となっている。

しかしそれでもなお、Smalltalk-80 はより一層の高速化が望まれている。専用ハードウェアの研究が盛んなのもこのためである。

さて最近ネットワーク透明な x-window システムが登場した。これによって離れた場所の計算機資源を手元のビットマップ端末で使う可能性がひらけてきた。例えば Smalltalk-80 実行専用ハードウェアをネットワーク上で共有して使う可能性がでてきたわけである。この可能性の実験として、菊32v の I/O システムの部分を X-window のプリミティブで置き換えてパフォーマンスを調べてみた。これはこの実験の報告である。

2. X-window システム

X-window は MIT で開発され、最近業界標準の有力候補となっているウィンドウ・システムである。TCP/IP プロトコルとソケットによるプロセス間通信をサポートする、Unix システムの上で使用可能である。このシステムの最大の特長はネットワーク透明であることである。

これによって、ホスト（クライアント）は

1. ビットマップを持つリモートマシン（サーバー）の上にビットマップを転送できる。

2. リモートマシンのキーとマウスの入力を取ることができる。

この機能のためのプロトコルがネットワーク上で転送され、このプロトコルを解釈するディスプレイ・サーバーがリモートマシンの上で動いているわけである。この機能により Smalltalk は、その I/O の部分を X-window によって置き換えることが可能となるわけ

ある。

なお、インプリメントに用いた X-window システムは、バージョン 10 である。従って、この報告の中で言及するシステムはこの版である。

3. ビットマップ・ディスプレイ

ビットマップのインプリメンテーションは、少し回りくどいものとなっている。それは以下の理由による。

1. Smalltalk-80 では、ビットマップは主記憶を直接ディスプレイにマップできるようなアーキテクチャを仮定している。X-プロトコルでは残念ながらこれをサポートしていない。

2. X-プロトコルでは ビットマップの操作として XBitmapPut がある。これは、ソースとディスティネーション（ディスプレイ）の間の16種のビットマップ演算 (2^2^2) をサポートするものである。これに対して Smalltalk-80 では、ビット・ブリティ・オペレーションと呼ばれる (src AND mask) OP mask という操作が要求される。mask は 16x16 のビットマップに限定されたそのパターンを繰り返し用いる特殊なものとなっている。また、クリッピングも必要である。つまり直接ビット・ブリティ・オペレーションをインプリメントするにはいろいろな機能が X-window には足りない。

3. Smalltalk-80 はビットマップのイメージを16ビットワードで描く。これは Dolphine, Dorado が 16ビットワードのワードマシンであったためである。バイトのオーダーはハイ、ローとなっている。しかし、X-プロトコルはバイトオーダーをロー、ハイを仮定している。（これは、X-プロトコルが Vax の上で開発された名残である。）そこでそのままビットマップを転送するとバイトオーダーの逆転した奇妙な図形となる。この補正をプログラムで行わなくてはならない。

そこで、実際のインプリメンテーションは以下の通りとなった。

- a) ディスプレイのイメージはオリジナルと同じ様に主記憶上におく。

b) ラスター・オペレーションが実行されるたびにディスプレイに書き込まれたかのチェックを行う。書き込まれていれば、その部分を X-プロトコルによりウィンドウに転送する。

この方法の利点はまさに書き直された部分のみのデータ転送となるので、ネットワーク上のデータ転送量が最小となることである。

欠点はクライアントの CPU を消費し、またもしサーバーのディスプレイに特殊なハードウェアがついていたとしてもこれを生かせないことである。

何れにしても、Smalltalk-80 のラスター・オペレーションは複雑な操作でこれを X-プロトコルに翻訳するのはかなり大変なプログラムである。

4. キーボード及びマウス入力

キーボードの完全なインプリメントも難しい（できない）。これは、Smalltalk は、全てのキーの押されたことと離されたことの情報を要求するためである。ところが、X-システムではコントロール・キーとシフト・キーの押し離しが検出できない。そこで、Smalltalk の CTRL-0 コマンド（フォントの切り替え）などが使えないこととなる。これは、X-プロトコルの不備とは言えまい。と言うのは、キーボードによっては、コントロール・キーや、シフト・キーの動作をハードウェアとしてサポートせず、結果のコードのみを出力するものがあるためである。

マウスは容易にインプリメントできる。マウスの押し離しと位置は X-システムで提供するキューに入っているので、これを Smalltalk インタプリタが適当なタイマー割り込みで取り込めばよい。X-システムではマウスはキーと違って押し離しをそれぞれイベントとして取り込める。これは、Smalltalk の要求にぴったりとあっている。

5. 評価

各種の X-ウィンドウの動くマシンで性能評価を行って

みた。クライアントにはすべて、Sun3/260を用いた。サーバーとして次の3種のマシンを用いた。

1. News nw-830
2. micro-vax-ii GPX
3. Sun3/260 HM

この性能評価を表1に示す。Sun3 上での直接のインプリメンテーションである 菊 32v を 100 としてその X-ウィンドウ版の性能を示した。これは、Smalltalk に含まれる標準ベンチマークの一部である。BitBLT は、ビット・ブリット・オペレーションの生の呼出しのテストであり、format text は、プログラムをディスプレイ上に表示するというディスプレイの性能のマクロな評価である。

裸のバージョンとの性能の差は歴然で少なくとも 30 倍程度の差がある。またネットワークの介在に関係がないことから、イーサネットの転送ネックではない。つまりオーヴァーヘッドの大部分は X-プロトコルから来ていることになる。ただ書換え面積がほぼ全画面（1000x1000）程度になればデータ転送量は 1M ビットにもなるからピーク 10M ビット/秒のイーサネットにとっても無視できる量ではない。

マクロなベンチマークではディスプレイそのものの遅さはそれほど関係がなくなり、どれも 2 倍程度の性能の低下となる。

しかし、ディスプレイの速さはユーザーインターフェースに重大な意味を持っており、速度が半分になることは容認しがたい。またネットワーク上をデータが転送されるため、こみ具合によってよってレスポンスが変わってくるのも気持ちの悪いものである。そうは言っても絶対的な性能は十分とは言えないまでもほぼ仕様可能な範囲内にはある。

	BitBLT	textDisplay
Sun3 original	0.6	1.70
Sun3/260HM(remote)	19.6	3.74
Sun3/260HM(self)	19.8	3.90
News-830	23.8	3.90
μ vaxII-GPX	26.3	3.86

表1. ディスプレイの性能

キーボードやマウスの性能を評価することは、意味がないであろう。人間が使うものである以上、必要以上に性能があっても意味がないためである。その意味で X-システムの能力には全く不足がない。

6. 結論

まだチューニングは十分ではなく、結論を出すのは早すぎる。しかし、この経験から

1. ネットワークに即時性がないことは、ユーザーインターフェースに悪い影響を与える。
2. 高速の Smalltalk クライアントがあれば、ネットワークの高速化と X-プロトコルの改良によってかなりの性能向上が見込める。

ということが結論できる。

謝辞

この様な異機種間結合の実験を可能とする ETL-1an の維持・管理を行って下さる電総研情報部門の皆さんに深く感謝します。

また、このインプリメントをするにあたって、菊 32v の共同開発者であった、寺田実氏（東京大学）のコードは大変参考になりました。また Smalltalk を X-ウィンドウに移植した横手靖彦氏（慶応大学）には、X-ウィンドウの部分のコードをネットワーク・メールで送って頂きました。当然のことながらこのコードは大変参考になり、このインプリメントでも一部はそのコピーを使わせて頂いています。この両氏に深く感謝いたします。

参考文献

- [1] Goldberg, A. and Robson, D. : Smalltalk-80 The Language and Its Implementation. Addison-Wesley (1983)
- [2] Scheifler, R. and Gettys, J. : The X Window System, 1986
- [3] Gettys, J., Newman, R. and Fera, T. D. : Xlib - C Language X Interface Protocol Version 10, 1986

```

etlsui 48% cd st180
etlsui 49% ls
Smalltalk-80.changes goodie src
Smalltalk-80.sources smalltalk xsma11talk
distribution snapshot.im xsrc
etlsui 50% xsma11talk -lm snapshot.im etlsui:0 &
[1] 797
etlsui 51% Displaw. etlsui:0

```

```

Image: snapshot.i
Mem: 0x3000e(10nq
OpsLeft = 2338
CoreLeft = 33ff2
^Z

```

```

etlsui 51% bg
[1] xsma11talk
etlsui 52% uim &
[2] 798
etlsui 53% screen

```

I

System Transcript
 Snapshot at: (22 November
 1985 10:16:00 pm)

System Browser

- Numeric-Magnitudes
- Numeric-Numbers
- Collections-Abstract
- Collections-Ordered
- Collections-Sequential
- Collections-Text
- Collections-Arrayed
- Collections-Stream
- Collections-Support

System Workspace
 The Smalltalk-80m System Version 2
 Copyright (C) 1983 Xerox Corp.
 All rights reserved.

Create File System
 UnlinkFile Initialize.
 Disk +
 (UnlinkFile)directory: nil directoryName: ''
 open.
 SourceFiles + Array new: 2.
 SourceFiles at: 1 put:
 (FileStream ofFileName: 'Smalltalk-80.sources').
 SourceFiles at: 2 put:
 (FileStream ofFileName: 'Smalltalk-80.changes').
 (sourceFiles at: 1) readOn.
 SourceFiles + Disk + nil

