The LBX Mini-HOWTO

Paul D. Smith, *psmith@baynetworks.com* v1.04, 11 December 1997

LBX (Low Bandwidth X) 是一種在 X Protocol 上進行壓縮的 X Server 延伸。他的用意在於改善低速網路連接時，X 程式與 X Server 顯示及回應的時間。

目錄

1 內容

X protocol 可能製造網路上的壅塞， 尤其是一些看似簡單的事情，例如開新的視窗。 任何人嘗試著在 28.8 或是更快的撥接網路上使用 X 開啓一個 X 視窗都能造成極痛苦的等待。

LBX 基本上是為了減少兩個系統的 X 訊息交換壓縮極快取的專案。

2 LBX 目前的狀況？

LBX 完全延伸在 1996十二月由 X 社群釋出 X11R6.3 的 X Protocol。 以 XFree86 家族而言，她算是 XFree86 3.3.x 系列的一分子。

3 誰能再 LBX 上獲得利益？

如果你利用撥接使用服務，LBX 將可以加速 X 程式於遠端執行，而在 DISPLAY 環境變數設定的本端的 連接。此外如果 DISPLAY 設定跨越 WAN (例如跨國的) 或是其他的低速連接， LBX 都有所助益。

4 誰不需要 LBX？

如果你鏡靜止需要在本端使用X 或是你根本不使用 X，當然， LBX 不會有任何助益。

如果你在高速的 LAN 上， LBX 也不會有太大助益。 有些人也許會問"如果 LBX 能減低網路交通，那麼為什會在 高速的 LAN 上沒有幫助？" 如果你的目的是在於減少網路流量 那麼它的確可能有所幫助 。 但是你的目的是減少 X 的回應時間， LBX 可能不是你所需要的。 雖然它使用快取以及 壓縮，但是他們也造成一些花費 (快取需要記憶體 壓縮需要 CPU 時間)。 如果你的連線已經很快速了 LBX 反而可能降低速度。

5 如何讓 LBX 工作？

LBX 利用在 X client 端的一個處理快取和壓縮的代理伺服器工作。 X 伺服器知道遠端使用這一個代理裝置，並依據他而進行解壓縮。

以下是 X clients的一般設定。在我們的討論中本端 (LOCAL) 指的是在你面前的工作站，你能見到它的螢幕。而遠端 (REMOTE) 指的是你遠端跑程式的工作站。

```
         REMOTE                                    LOCAL
      +-----+                                                    +-----+
      | APP |-\            Network             +----------+  |     |\
      +-----+  \---------------------------->| X SERVER |=>|     ||
      +-----+  /       (X Protocol)          +----------+  +-----+\
      | APP |-/                                                 /_____//
      +-----+
```

當使用 LBX，一個代理伺服器 (`lbxproxy`) 在遠端被處理，現在本端先和代理伺服器溝通，而不是直接和程式溝通。 接著這程序處理快取以及壓縮 X 的要求並傳遞他們。 他們看起來像是：

```
          REMOTE                                              LOCAL
                                                                        +-----+
      +-----+  +-------+           Network            +----------+  |     |\
      | APP |->| PROXY |------------------------------>| X SERVER |=>|     ||
      +-----+  +-------+       (LBX/X Protocol)       +----------+  +-----+\
      +-----+   /                                                      /_____//
      | APP |--/
      +-----+
```

至於到底是什麼東西被壓縮以及被快取則不是這放文件討論的範圍。

6 我需準備要什麼去使用 LBX？

你需要一個已經將 LBX 編譯整合在本端的 X Server。 除非你在編譯的時候指定不將 LBX 整合於其中， 否則 X11R6.3 會自動的使用 LBX 的特性。 所有的 XFree86 3.3 server 預設都會把 LBX 整合於其中。

你可以使用 `xdpyinfo` 這個命令檢是你的 server 是否打開了 LBX 的功能。 執行 `xdpyinfo` 即可在 number of extensions 的列表下面，見到 LBX 列在其中。

下一不你需要一個在遠端已經將 `lbxproxy` 這程式編譯在內的系統。 這邊有個陷阱。 如果遠端的系統和你本端不同，那麼在本端的 `lbxproxy` 將不會作用。

很不幸的，沒所所謂分離開的 `lbxproxy`，所以 你必須 (a) 所有的遠端系統都使用X11R6.3，或是 (b) 找一個針對你的本端系統預先編譯好的 `lbxproxy`。 當然後者是比較方便的。

`lbxproxy` 僅僅是一個單一執行檔。 它沒有任何的設定檔或是資源檔，或諸如此類的檔案。

7 我不需要準備什麼去使用 LBX？

遠端系統不需要一個新的 X server (如同一般的狀況，遠端系統並不需要任何一個 X Server).

你所需的程式並不需要指定特別版本的 X 或是特別的函式庫。 我借由 LBX 使用 X11R5 的程式，也沒碰到什麼問題。

你不需要使用 root 或是特別的權限存取選端系統。 你只需要一般的權限執行 `lbxproxy`。 另外你只需要在你自己的家目錄下執行即可，lbxproxy 並不需要安裝在特定的目錄。

8 如何使 LBX 開始工作？

OK，接下來的步驟都很簡單。 將下列 LOCAL 以及 REMOTE 用你本端以及遠端的主機名稱取代。 (千萬不要搞混了)On LOCAL:

1. 啓動你的 X server。
2. 告訴你的 X server 哪些遠端系統是被允許存取的。 使用 host-list 的方法，鍵入 `xhost + REMOTE` 。如果你要使用 `xauth` 你需要作更多事情；請參閱 man *xauth(1)* 以獲得更多的資訊。

如果你不熟悉 X 遠端的存取，那麼你需要參閱 *Remote X Apps Mini-HOWTO* `<http://www.xs4all.nl/~zweije/xauth.html>`。

On REMOTE:

1. 起始 `lbxproxy` 並告訴它傳送訊息到本端的 X server，如下：

   ```
   $ lbxproxy -display LOCAL:0 :1 &
   ```

   這告訴 `lbxproxy` 在遠端系統使用 display `:1`；如果系統大於 1 的 display 已經在使用。你可以使用 `:2` 之類的設定取代。

2. 設定你的 DISPLAY 環境變數，使此環境變數和你 `lbxproxy` 所設定的相同。取代平常的 DISPLAY 設定

   ```
   $ DISPLAY=:1
   $ export DISPLAY
   ```

   如果你使用 csh 之類的，以下可能是你所需要的：

   ```
   % setenv DISPLAY :1
   ```

3. 如果你使用 `xauth` 你需要確定你的 cookie 正確的運作。你可以從 *Remote X Apps Mini-HOWTO* `<http://www.xs4all.nl/~zweije/xauth.html>` 獲的更多的資訊。

4. 執行你的 X 程式！

接下來就完成了；所有在 Display 環境變數為`:1` 的 X 程式都會在都會使用 LBX。當然，你沒有辦法起始一個 X 程式，其 Display 的環境變數是指向 `LOCAL:0` 以及在同一個時間下執行。

9 問題

這邊有一些常見的問題：

Q)

`lbxproxy` 以 access denied 的錯誤結束程式。

A)

這是代表你進端的系統不接受選端系統的連結而產生的錯誤。 請參閱 *Remote X Apps Mini-HOWTO* `<http://www.xs4all.nl/~zweije/xauth.html>` 有更詳盡的說明。

碰到這問題時，有一個簡單的試驗方式。請試著在遠端不要透過 `lbxproxy` 執行一個簡單的 X 程式， 比如說 `xclock` ：

```
$ xclock -display LOCAL:0
```

如果它不能正常的工作，那麼就表示是`xhost`或是 其他一些基本的 X 問題而不是 LBX。

10 Documentation

在標準的 X 文件中，唯一可以參閱的文件應該是 man *lbxproxy(1)*。

如果你有看 X 的原始檔，那麼下列有些有趣的資訊有關 LBX ：

- `xc/doc/specs/Xext/lbx.mif` (Framemaker MIF)
- `xc/doc/hardcopy/Xext/lbx.PS.Z` (Compressed Postscript)
- `xc/doc/hardcopy/Xext/lbxTOC.html` (HTML)

LBX 更深入的演算法在下列的地方：

- `xc/doc/specs/Xext/lbxalg.mif` (Framemaker MIF)
- `xc/doc/specs/Xext/lbxalg.PS.Z` (Compressed Postscript)

如果你沒有 X 的原始檔，你可以由 *X* 組織的 *FTP* 站取得 `<ftp://ftp.x.org/pub/R6.3/xc/doc/>`。

11 Alternatives

如果你有某些原因不喜歡 `lbxproxy`。例如你不喜歡他的執行效率， 它不幫你工作， 你不想麻煩的為遠端建立 lbxproxy 或著你只是想是是其他的選擇， 這邊有一些其他的 X protocol 壓縮方式。 (有沒有人有其他的方案？)

11.1 dxpc - 一個極不一樣的 X Protocol 壓縮程式

- 原本的作者： *Brian Pane <brianp@cnet.com>*
- 現下的維護者： *Zachary Vonler <lightborn@mail.utexas.edu>*

dxpc `<http://ccwf.cc.utexas.edu/~zvonler/dxpc/>` 和 LBX 以相同的方式運作。然而為了避免修改 X Server的原始碼 以及實作一個 X Server 的擴充`dxpc` 使用 兩個 代理伺服器： 一個像是 `lbxproxy` 跑在遠端的系統，另一個則在本端執行。

在遠端的代理伺服器負責溝通 X Client 以及本端的代理伺服器， 而本端的代理伺服器負責溝通 X Server 以及遠端的代理伺服器。

所以，對於 X clients 以及 X server 兩造，他們都以平常的處理方式處理 X Protocol

11.1.1 優點

- 既然它是完全的獨立程式而不需要動到 X 的內部， 它一定非常的 容易編譯以及安裝。
- 它是獨立的被維護著，所以你不需要 OSF 釋出新的 X，即可過的效能的增進以及錯誤的修正。
- 它提供了比 `lbxproxy` 更多的壓縮資訊以及統計資訊。

11.1.2 缺點

- 它不是 X 的標準套件，你必須自己取得以及編譯安裝。
- 它在安裝上變的比較複雜，因為它要求在本端也要有一個像是遠端的代理伺服器。

11.1.3 我在哪裡可以得到 dxpc？

dxpc 的原始檔可以自*ftp.x.org* <ftp://ftp.x.org/contrib/utilities/>取得。

這裡有一個網頁 <http://ccwf.cc.utexas.edu/~zvonler/dxpc/>有 dxpc 很多不錯的資訊。包含了 dxpc 的 mailing list，原始檔，以及 為多種平台預先編譯好的執行檔。

11.2 Ssh (Secure Shell)

Ken Chase <*lbxhowto@sizone.org*> notes that *ssh* <http://www.cs.hut.fi/ssh/> 能被用於壓縮。雖然它只要的目的是為了保密，但是它也壓縮它傳送的資料。

所以如果你自一個 `ssh` 連結執行 X，你在使用間自然會壓縮資料。

11.3 誰比較好？

我不知道。 LBX 以及 `dxpc` 在純壓縮上比 `ssh` 好。當然， `ssh` 提供了在安全上的附加功能。 不過，也沒有任何理由禁止你使用 `ssh` 加上另外兩種之一的壓縮方式，以獲得良好的壓縮以及安全性。

在這些選擇上跑一些評量以獲取統計資訊，應該不是很難的事。 但是我沒有做過這件事，也沒有聽說過任何人做過。