

Remote Forwarding of Human-Machine Interfaces for Industrial Controls

— CERN Summer Student Report —

Pierre WULLES*

Supervisor: Dr. Marc BENGULESCU[†]

^{*,†}CERN, BE-ICS-FD

August 9, 2018

1 Introduction

At CERN, people need sometimes to access graphical application through the network. Indeed sometimes it is not possible to install the software that you need or maybe your computer is not powerful enough to use some applications used at CERN. There are also applications that are used to monitor the critical part of CERN and these are running at all times. Until now, Windows Terminal Servers was used as a way of screen forwarding applications, but it has several issues regarding simultaneous access of a resource and the drawback that licenses are expensive. Therefore we should think about moving to something less costly and preferably open-source.

This is where I come in: the BE-ICS group has already thought of using Xpra: X Persistent Remote Applications, an open-source software that allows you to access distant running software. My goal was to benchmark this software to see if it is able to do as much as Windows Terminal Servers used to do and maybe even more! To do so I built a matrix of tests to know which applications are running well and which are not. Then I benchmarked the software used in the BE-ICS group: WinCC OA, a software made for designing industrial control systems. Eventually I managed to setup a session of Xpra through a proxy. In what

*pierre.wulles@cern.ch

[†]marc.bengulescu@cern.ch

follows, I will try to summarize my efforts in these activities. A more detailed technical report is available from the BE-ICS-FD section.

2 Testing with mainstream software

First and foremost, I spent some time going over the documentation to familiarize myself with the numerous options that Xpra offers and then I tried Xpra, by running a server and a client on the same Linux computer with a simple graphical application: `xterm`. It worked well both on the native client and the browser client. Indeed, one of the great feature of Xpra is the possibility for the user to access graphical application within the browser, without installing anything.

After this first success, I've tried a lot of application such as Emacs, LibreOffice, Inkscape or even intensive graphics application as Blender. Most of them were working fine. I only noticed some minors glitches and in a general manner: saving a file is quite long. It was working on both browser and native clients even though I noticed that the browser version suffers from more bugs than the native client. I also spotted a weird bug about the keyboard, the key '>' is forwarded as a '<'. Also for some applications, the graphical interface is slightly modified, buttons don't appear exactly with the same color, but it is not a major problem at all.

At this point I started testing applications on two computers by setting up a virtual machine (VM). I've also tested the client on Windows. The results are: on two computers it worked as well as it worked before except that sometimes I've experienced minor lags. On Windows, a lot of application were suffering of aliasing problems, I suspect the graphic library Gtk to be responsible of this. But it was not a major problem. Notice also that on Windows users can use a graphical interface to connect to an Xpra session.

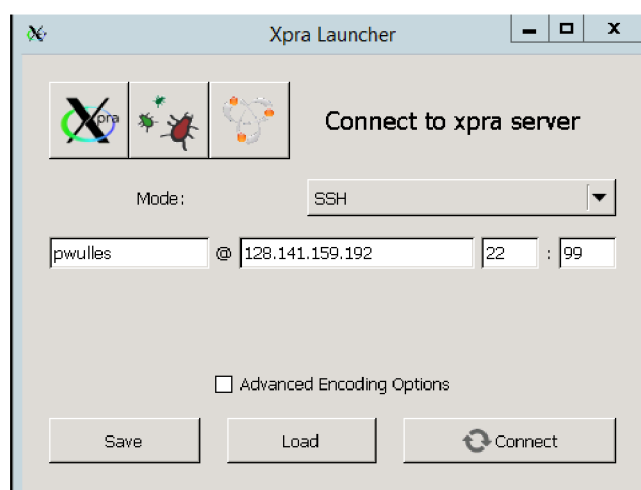


Figure 1: Windows client for Xpra

It is also possible to screen forward a whole desktop with Xpra. I have tested this functionality and it works as far as you don't want to forward Gnome or KDE! However I succeeded with fluxbox and openbox.

I want also to draw your attention that it is not recommended to use two session of Xpra at the same time on the same client or using another application that needs a lot of bandwidth, such as Skype.

For a basic use of Xpra as I did here you can try:

- `xpra start :100 --start=xterm #On the server`
- `xpra attach :100 #On the client`

Of course here we assume that both client and server are running on the same machine. More complete examples are present in the technical report.

3 Testing with WinCC OA

The BE-ICS group wanted to be sure that the software that they use to develop SCADA applications, WinCC OA, is fully supported by Xpra. So as I did in the previous part, I built an application that employs the most used graphical functions of WinCC OA. I have tested functions related to opening new panels, login dialogs, password dialogs, hiding or showing layers, etc. Everything was working as expected, but for one thing: modal windows. A modal window is a window that stays on the top and prevents actions in other windows. The window was popping up all right but it was still possible for the user to ignore it and continue to do things in different windows.

In the purpose of fixing this issue or at least to understand where it came from, I've contacted the developer. First he answered me a few hours later that I haven't provided enough information, so I sent a more complete email with version of Xpra, operating system and log file. He replied back to me that he had added the functionality in the new version, all this discussion has been reported in a ticket on the Xpra website. After this I had a new reason to think using Xpra was a good idea as the developer was so responsive and wiling to help.

Then I've started trying to use the UNICOS framework. It is a framework that helps the development of control applications concerning the technical infrastructure, such as monitoring cryogenic systems and other aspect of machines running at CERN. The framework was working well and I didn't notice any particular bugs except one, that was really important but also strange. When trying to open a login dialog, the pop up appears for a few milliseconds and then disappears. But the weirdest thing is that it doesn't happen every time, sometimes the window stays open. After a meticulous investigation, we noticed that the WinCC OA

function: `StayOnTop()` was responsible for this! I've tried to reproduce the bug only with Qt but I didn't succeed, so the developer was not able to help me more as he could not reproduce the bug. But if it is one day possible to send him, by asking maybe Siemens – the developer of WinCC OA, how the function `StayOnTop()` really works then he could probably be able to fix this issue. The good point of this is that my supervisor has found a way to work around with a little script that can catch the disappearing window and put it back on the screen.

Except for this bug, which is more or less fixed and probably totally fixable if we can get some feedback from the WinCC OA developers, the test of WinCC OA was quite positive.

4 Testing with proxy

One of the main purpose of Xpra is to be used as a way of starting a session remotely. We want to have an architecture with server, proxy and client, this will allow a client to connect from the general purpose network (GPN) to the technical network (TN) through a gateway secured with authentication.

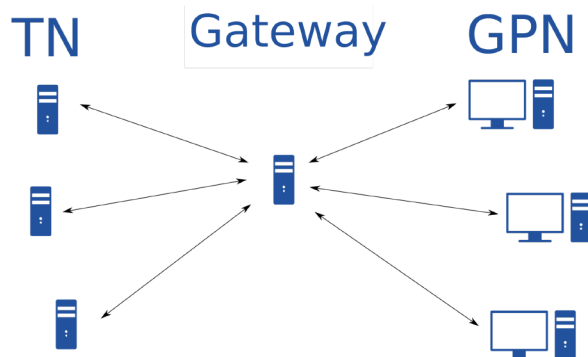


Figure 2: Architecture needed

So first I started to try this kind of architecture on the same machine and it was working as expected: I could attach to the server. Then I tried with two and eventually three machines; here I've spent a lot of time on issues related to the firewall, which I finally ended up disabling. Another problem that I encountered was the authentication: it is not possible with Xpra to send a password without encryption, i.e. the mode `allow` cannot work when trying to do the connection via proxy on distinct computers. So after I realized this, thanks to the developer's feedback, I used a password stored in a sqlite database and it worked.

But what we wanted to do with Xpra is to *start* a session remotely, not only attach to it.

So I've tried to do it with a lot of different ways, it didn't work. I then asked the developer for help. He replied back almost one week later that this functionality didn't exist and that he has implemented it for the latest beta version of Xpra. I've installed and tried this beta version but I encountered some bugs related to pygtk so I was no able to properly set up the proxy. This was done using the very last version of Xpra that I have tested, i.e. Xpra **v2.4-r20056**. This is also where my summer student program comes to an end, but I am confident that it is only a matter of time until this issue will be fixed. Given the developer's short reaction time and enthusiasm for Xpra, for sure in the next beta release everything will be working properly.

5 Conclusion

To conclude I would like to answer to the question that was initially asked to me: Is it possible to use Xpra in production at CERN, and if yes what should we take care of? My answer is clearly yes, even though there are some minor bugs that could be annoying or functionality that is not completely operational. The commitment and dedication of Antoine Martin, developer of Xpra, is something to take into account. He has always answered my questions, even though I was not always really clear in what I asked, he always wanted to help and has several times implemented functions that I was asking for. Xpra is an open-source project but not an abandoned one as we can often see, this one is lead by someone invested and skillful. Xpra and particularly the next version, will be quite complete for an usage at CERN. I have created many JIRA tickets on Xpra on how to correct minors bugs that I spotted and usual things such as installing Xpra, upgrading it, etc.

Notice also that this is the reduced version of a comprehensive Technical Report made for the BE-ICS group that contains all the results of the matrix test and a more complete inventory (but still not exhaustive) of encodings and options available in Xpra.