

X2go Remote Desktop Solution Evaluation Report

2015

Version 1.5



Table of Contents

- [A. Document History](#)
- [B. Document Scope](#)
- [C. Executive Summary](#)
- [D. Introduction](#)
- [E. Prerequisites](#)
- [F. Evaluation Details](#)
- [G. Suggestions for Evaluation Procedure Revisions](#)
- [H. Evaluation Result](#)

A. Document History

Document History				
Entire Document	0.1	06/26/2015	Created Document	S. Litzinger
Prerequisites, Evaluation Details	1.0	07/15/2015	Edits	Team
Test Steps	1.1	07/22/2015	Configure for X2go	Team
Requirement	1.2	07/30/2015	Evaluation details	Vana
Results	1.3	08/21/2015	Results	Susan
Evaluation Results	1.4	09/10/2015	Evaluation Results	Vana
Entire document	1.5	09/16/2015	Group revision	Team

B. Document Scope

This document is the evaluation report for the functional evaluation of X2go. The results of the evaluation and the collected information here are intended for the benefit of the XSEDE project and the general eScience community.

C. Executive Summary and Recommendation

X2go was selected for evaluation because campus bridging users want the ability to open and maintain an interactive graphical session on a remote resource for a period of at least several days. There is also a need to be able to connect multiple users to the same server at the same time with each user having their own, individual desktop session. In addition, desktop sharing (Remote Assistance) is desirable.

The evaluation was performed on the open source version of X2go, which is a remote desktop program that allows a user to interact with a machine as if they were sitting in front of it. X2go client version 4.0.4.0 and server version 4.0.1 were evaluated.

In general, X2go worked well allowing multiple users to connect to a single server each with their own, independent desktop. However, long duration connections were a problem. While some client connections stayed open for an entire week, most of them failed after a day or two. When the connections failed, they often did so without reporting a cause. These connections can be resumed from the client so there is a workaround for the issue.

We recommend the open source version of X2go with some reservations. It will meet the user's requirements but not without some inconvenience.

D. Introduction

The members of the evaluation team were Peter Enstrom, Susan Litzinger and Vanamala Venkataswamy. The evaluation was started in June 2015. The evaluation finished in August 2015.

X2go provides a remote desktop solution that allows users to view and control their desktop from remote locations. It allows users to run their graphical applications on a remote machine's desktop through their X2go client using any reasonable network connection. It uses the NX technology to compress high level X11 calls allowing it to perform well even over low bandwidth connections.

X2go client is available for all variations of Linux as well as MacOS and Windows. This evaluation covers the open source version of X2go. X2go server is only available for Linux.

X2go is under active development and a version of the client has been released since this evaluation concluded. This new release fixes some issues with Windows including a bug where a session that started under the Windows client could not be resumed under Linux.

There isn't a separate enterprise version of X2go, however the X2go team offers various aspects of professional support to users, such as workshops, training, feature development and integration.

E. Prerequisites

The only daemon/service that the X2go server needs to be running is the SSH daemon. If users can connect to the server via an SSH client, then they should be able to connect via the X2go Client.

On some Linux distribution, there may be an X2go server init script called X2goserver or a systemd service called X2gocleansessions.service. It is highly recommended that this service be enabled. It just handles session cleanup and does not need to be running in order for the client to connect to the server.

F. Evaluation Details

The requirements for this evaluation were:

1. Run for at least 1 week uninterrupted
2. Have a very high degree of reliability for maintaining open sessions
3. Support audio and video
4. Maintain a responsive, steady network connection
5. Update the screen reasonably quickly
6. Low latency mouse tracking and scrolling
7. Support dual monitor configurations
8. Allow file transfers between the server and client machine
9. Support single user, multiple sessions to same server
10. Support multiple users, concurrent login

In order to evaluate the software a test with four parts was created that detailed the actions for the evaluators. The results of their testing were captured in test logs that are linked on the wiki page for this evaluation. The first part tested the basic functionality of connecting to a remote desktop, viewing and controlling it. The second part of the test exercises X2go's more advanced features. The third part of the test introduced faults to see how X2go would handle them. The last part of the test investigated X2go's ability to maintain a long term connection to a desktop.

Testing took place on servers at NCSA, PSC and UVa. In order to test connections from a remote network, home machines were also used. A template virtual machine (VM) containing X2go was created on the NCSA OpenStack cluster. Virtual images on the cluster were created from the template and used in the testing as well. Multiple operating systems were used in testing: Windows, MacOS, and various Linux flavors (CentOS, SuSe, and Ubuntu). The remote desktops used for the test included basic desktops along with servers with high resolution and dual screens.

Test 1 : Basic Functionality

In this section we tested routine actions a user would expect to be able to perform on their desktop. In order to test how responsive the update of the desktop was, we used a browser to display video with audio. This allowed us to easily detect any sluggishness in the output. We also tested cut and paste from the X2go window to a native window (and vice versa) on the client machine. Also tested was X2go's feature that allows files to be passed between the server and the client. Each of these tasks were repeated using clients and servers running on different operating systems.

Steps for test 1

1. Select 4 different test machines, different types, different OSs.
2. Create accounts for 2 users, "tis" and "tis2".
3. Install X2go client/server on machines A and B which are on the same LAN.
4. Install client only on machine C which is on the same LAN.
5. Install client only on machine D which is outside the LAN.

6. Connect from A to B as user tis, pass a file from A to B, retrieve a file from B to A, open a browser and play video with sound, cut and paste text from the X2go window to a native window, disconnect.
7. Connect from C to B as user tis, pass a file from C to B, retrieve a file from B to C, open a browser and play video with sound, disconnect.
8. Connect from D to B as user tis, pass a file from D to B, retrieve a file from B to D, open a browser and play video with sound, disconnect.
9. Connect from A to B as user tis, connect from C to B as user tis, disconnect.
10. Connect from A to B as user tis, connect from C to B as user tis2, disconnect.
11. Edit a document

Results

File transfer between machines on the same LAN and between machines on different LANs all worked as expected. Transfer rates were reasonable between machines on different LANs. There were no noticeable speedups or delays when files were transferred between client/server machines on the same LAN and client/server machines on different LANs.

Copying and pasting text between client/server machines worked seamlessly.

Playing video on a server machine and viewing it on a client also worked. Mostly the sound worked well. There was a case where the sound was heavily buffered which rendered it unusable. This cleared up when the test was re-run on the client after the X2go session files were deleted and the client was re-installed.

When a second client connection takes over a server desktop that another client is currently connected to, the first client is disconnected as the second connection is established as expected. No warning is given to the first client and they are not given the option to either accept or reject the second connection.

Connecting to the same server machine from two different client machines as different users worked. The connection from machine A to machine B as user tis remained active as expected when user tis2 connected to machine B, and the new connection from C to B was established with its own, separate desktop.

Test 2 : Advanced Functionality

In this section, we tested features beyond the basic set. By default a user establishes an X2go connection using a username and password. In this test an alternate login method was tested using private and public keys. The network was interrupted briefly to see how X2go handled the interruption. A longer interruption was then tested as well. We tested how X2go would react to a dual screen setup and a high resolution screen where more information is transferred. Also tested was how X2go reacted to chaining connections through more than one machine.

Steps for test 2

1. Select 4 different test machines (A, B, C and D), different types, different OSs.

2. Test machine C should have a dual screen setup.
3. Test machine D should have a high resolution screen (1900 x 1200 at least).
4. Create accounts for user "tis" on the machines.
5. Install X2go client/server on all 4 machines.
6. Connect from A to B as user tis using a private key, open a browser and play video with sound.
7. Briefly interrupt the network while connected, see if it recovers.
8. Interrupt the network for several minutes then restore.
9. Disconnect.
10. Connect from A to C as user tis, open a browser and play video with sound, disconnect.
11. Connect from A to D as user tis, open a browser and play video with sound, disconnect.
12. Connect from D to A as user tis, on machine A connect through no machine to machine B, open a browser and play video with sound, disconnect.

Results

X2go did not handle network interruptions very well. The connections failed when the network was interrupted for more than 5 minutes. A brief network interruption for one minute did pause the video but the video resumed when the connection was reestablished. An error message was displayed and the connection was terminated after the network connection was interrupted for 5 minutes or more. A new connection to the same desktop could later be established through the client. Panning between two screens on a dual screen server with each screen showing a video caused both the video and the mouse tracking to slow down but worked OK other than that. Chaining through two X2go connections did not cause any problems.

Test 3: Error Reporting

In this section we tested how X2go would handle induced error conditions and how it would report the problems to the user. We tested how X2go reacts when a second user connects to the same server that another user is already connected to. We tested switching between which remote user is connected to the server. We tested how X2go reacted to failed connection attempts due to trying to connect to a non-existent server, and by connecting to a server that is not running the X2go server software. We tested having two users try to control the desktop at the same time, one from the server's desktop and the other remotely from a client connected to that desktop through X2go.

Steps for test 3

1. Set up machine A using SUSE Linux.
2. Select 2 different test machines (B and C), different types, different OSs.
3. Create accounts for user "tis" on the machines.
4. Install X2go software on all 3 machines.
5. Repeatedly connect from machine B to machine A with X2go, log in, log out, repeat several times. When done with the test, disconnect.
6. Connect 2 different users to the same server from different client machines. Connect to the local desktop using the same user that is logged into the server desktop locally, try

again as a different user. Make connections from 2 different clients using 2 different logins, the 2 new desktops are independent of each other.

7. Try to connect from machine B to a non-existent machine. A meaningful error message should be displayed.
8. Terminate the X2go server on machine A. Attempt to connect from machine B to machine A. A meaningful error message should be displayed. Restart the server on A. Retry the connection, it should work. When done with the test, disconnect.
9. Connect from B to A. Attempt to use the mouse and keyboard both remotely from B and locally from A at the same time. Should work.
10. Reboot the server while an X2go session is active from a remote client.

Results

X2go handled switching between users well. Attempts to connect to an X2go server that failed gave adequate messages regarding the problem with the connection. Using the server's desktop while someone else is connected through X2go worked fine but was disconcerting if they weren't aware of the client control. There is no indication on the server desktop that there is an active, remote X2go client controlling it.

The Windows client sometimes could not suspend or terminate a connection. Stopping the X2go client would still terminate the connection. There were some cases where the Windows client would crash.

Test 4: Remote/Long-term Connection

In this test, we exercised long-distance, long-duration connections between clients and servers. Server machines were at NCSA and client machines were at UVa and PSC. Server machines were running YouTube videos in a loop to continually update the display to make it easy to verify that the connection was still active. Clients were connected to the servers and the connections were left alive for 7 days. We checked on the connections every day during the 7 days to make sure connections between the clients and servers were still active.

Steps for test 4

1. Select 4 different test machines (A, B, C and D), different types, different OSs.
2. Test machines A, B and C should be on the same LAN.
3. Test machine D should be off site.
4. Create accounts for user "tis" on the machines.
5. Install X2go client/server on all 4 machines.
6. Connect from A to B as user tis using a private key, open a browser and play video with sound. Leave the connection up for 7 days and check that the connection is still active daily during the test.
7. Connect from C to D as user tis using a private key, open a browser and play video with sound. Leave the connection up for 7 days and check that the connection is still active

daily during the test.

Results

Despite numerous attempts, we were rarely able to keep a connection up for 7 days. X2go seemed unable to maintain a stable connection for a week or more. Connections up to a couple days in length were pretty stable. No error messages were displayed and no log entries could be found to explain why the connections were dropped. The sessions could be resumed through the client without needing to restart the client.

G. Installation Information

Visit the X2go home download page (wiki.X2go.org/doku.php/download:start) to retrieve the software. The server is available only for Linux variations. The client is available for Linux, MacOS and Windows.

For X2go Linux client, use either apt-get or yum install, depending on your platform. For details, go to <http://wiki.X2go.org/doku.php/doc:installation:X2goclient>

For the X2go Linux server, use either apt-get or yum install, depending on your platform. For details, go to <http://wiki.X2go.org/doku.php/doc:installation:X2goserver>

For Windows client, download the [X2goclient-{version}-setup.exe](#) from this page <http://wiki.X2go.org/doku.php/doc:installation:X2goclient> and run the executable.

H. Usage Information

After installing the client or server, an icon shows up on your desktop on Windows machine. You just select the icon and the session starts.

In order to establish an X2go connection you first create a session through the client. You provide connection information for the server and also a connection type. There are a lot of different connection types available. We tested “connection to local desktop”, KDE, GNOME, XFCE (a free desktop environment), “internet browser” and terminal.

I. Evaluation Result

We recommend X2go for use connecting multiple users to a single server, each with their own independent desktop. It worked well allowing users to suspend sessions and either resume it from the same client or grab it on a different one. While it's performance was acceptable, some other remote desktop applications performed better for streaming videos.

If maintaining a long term connection to the server without interruption is important to the user, we would not recommend X2go since it tends to drop connections after couple of days without any error messages or useful information about the failure. The remote desktop was unaffected by these dropped connections and the connection from the client could easily be re-established so, while inconvenient, it was not an insurmountable problem.

The installation and user interface for X2go were well documented and easy to use. The project is currently under active development and the X2go team are quick to respond to user issues.

J. Suggestions for Evaluation Procedure Revisions

None