Performance Testing Citrix Applications Using LoadRunner:

Citrix Virtual User Best Practices

Elite Partner

hp

Software

Table of Contents

Introduction

In order to ease the burden of administration, save bandwidth, and improve user experience for response times, more and more companies are leveraging the Citrix technology platform. When moving applications to a thin environment, one of the difficulties is determining how the application will perform when multiple users are accessing it. HP addresses this by providing the Citrix recording protocol in LoadRunner. This white paper discusses the technical aspects of testing Citrix application with LoadRunner and offers a best practices guide to business process creation.

Citrix allows companies to publish traditionally thick-client applications and Windows desktops to end users in a thin-client manner. It allows users to run applications or thin desktop sessions using the resources of the Citrix server, as opposed to using the resources of the user's computer. When running a Citrix session (whether it be a published application or a thin desktop), the Citrix client on the local machine presents essentially a bitmap image to the user. When the user uses the mouse, it is sending x and y coordinates of the mouse activity to the Citrix server. The server translates the coordinates into mouse movement and sends updated bitmap images to the user's computer.

HP and Citrix have formed a technology alliance which allows testing Citrix applications using the native ICA protocol directly using LoadRunner. This allows the performance engineer to utilize the same methodology that applies to testing web and client/server applications in LoadRunner. Citrix is the second most popular technology when Northway Solutions Group services are requested. This white paper can be used to help organizations determine the appropriate starting point.

How It Works

LoadRunner is able to use the Citrix client to talk with the Citrix server and record the traffic going back and forth. The script that LoadRunner generates makes use of bitmap synchronizations. It is able to sync bitmaps based on a hash value that LoadRunner creates. A bitmap is an uncompressed image file that is a two dimensional array of pixel information. LoadRunner has an algorithm that takes this pixel information and creates a hash value based on this it. When the script is rerun, if the bitmap looks the same as before, the hash value will be the same, and thus the sync will work. Besides bitmaps, the LoadRunner Citrix protocol makes use of keyboard input and x and y coordinates (to know where to click or sync).

Citrix Scripting Best Practices

Because the Citrix protocol relies heavily on bitmaps and coordinates to work correctly, there are few things to keep in mind when recording and editing the scripts.

Display Settings

When writing Citrix scripts in LoadRunner, check the display settings of the generators where the scripts will be run. The color depth and resolution of the generators need to be the same as the settings on the computer where the scripts are written. If the settings are different, any sync_on_bitmap functions will fail. Along the same lines, avoid using Remote Desktop Connection when writing scripts or running scripts. When RDC connects to a computer, it changes the display settings that are in effect on the computer.

Keyboard Entry vs. the Mouse

During the recording of a business process, navigate the application being recorded using the keyboard instead of the mouse. When LoadRunner captures mouse clicks, it records the coordinates that were clicked on the screen. Replaying these coordinates is not very reliable. If your script relies heavily on coordinates, the scripts are more susceptible to being broken by

application changes. As opposed to mouse clicks, navigate the application using the tab key or shortcut keys. Even if the application changes later, the script can be easily fixed by adding or deleting tab key entries in the script.

ICA Files

ICA files contain connection information for Citrix applications. Double-clicking on an ICA file will launch the application. A Citrix administrator can create this file for the person writing the LoadRunner script.

When setting up connection information inside of LoadRunner, it is best to use an ICA file to connect. This makes the script more portable and more resilient to server change. If any of the connection information changes, in order to point the script to the new location, put the new ICA file in the script directory and replace the old one. For good housekeeping, it is a good idea of keep a copy of the ICA file that is currently in use in the root folder of the scripts folder. For example, if you are putting all of your scripts in c:\LoadRunner scripts\Project Name, you should put a copy of the ICA file in c:\LoadRunner scripts\Project Name.

The following is an example of what an ICA file might contain when opened in Notepad.

```
[WFClient]
Version=2
HttpBrowserAddress=citrix.app.us
TcpBrowserAddress=10.1.1.1
TcpBrowserAddress2=10.1.1.2
TcpBrowserAddress3=10.1.1.3

[ApplicationServers]
WEBS Load Testing=

[WEBS Load Testing]
BrowserProtocol=HTTPonTCP
Address=WEBS Load Testing
```

```
InitialProgram=#WEBS Load Testing
DesiredHRES=800
DesiredVRES=600
DesiredColor=2
TransportDriver=TCP/IP
WinStationDriver=ICA 3.0
```

Randomizing information

Since Citrix scripts contain a lot of bitmap and coordinate information, it can be difficult to randomize the input data. One of the places to randomize data is where drop down boxes are used in the business process. In order to do this, use the rand() function combined with the ctrx_key() function. The following is an example from a script where the value of the variable claim_type is randomized. This value could be used to search for a different type of claim object in a drop down list each time the script was executed.

```
random_number = rand() % 5+1;

claim_type = random_number;

for (counter = 1 ; counter <= claim_type; counter ++)

        ctrx_key("DOWN_ARROW_KEY", 0);
```
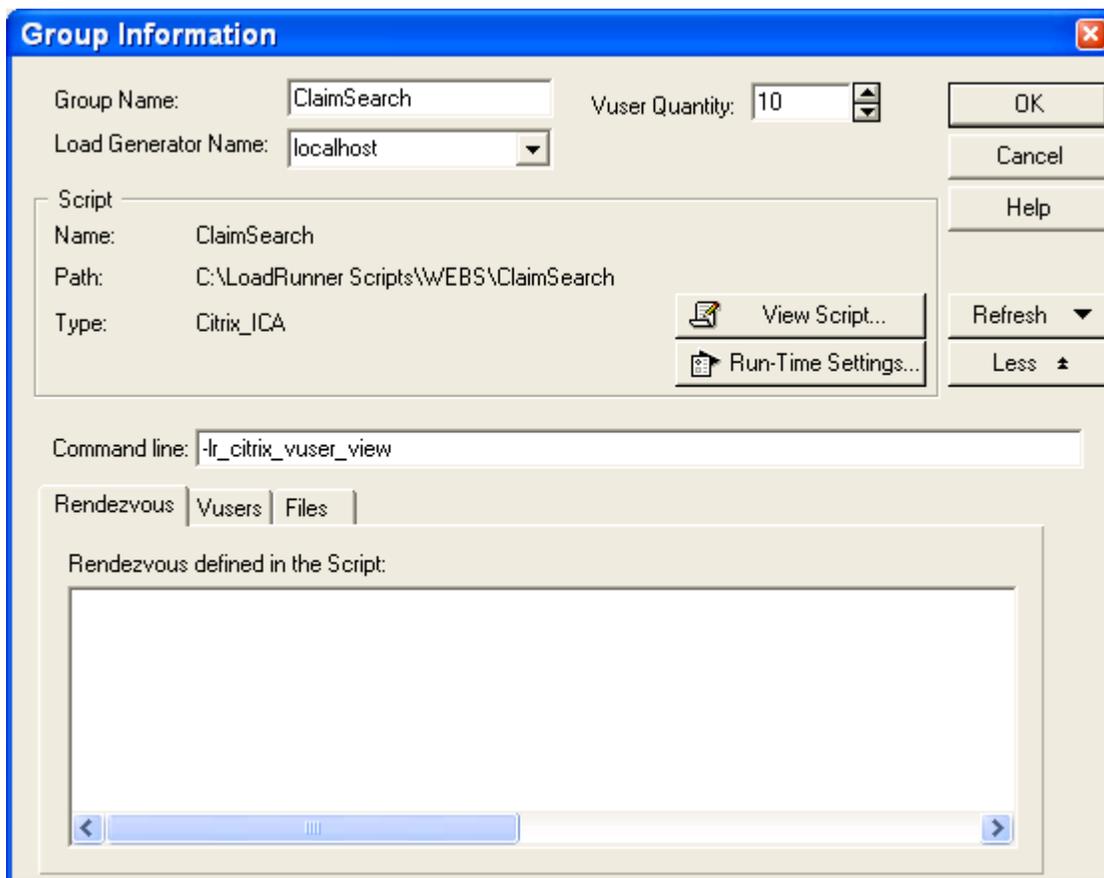
Debugging Vusers in the Controller

Sometimes Citrix scripts do not work the same on different computers. Most of the time it is because of problems like color depth. If the Citrix scripts have problems running in the Controller during the debug run, there is a way to see visually what the scripts are doing. In the Controller, bring up the details for the group that is having problems. Under the Details section, make sure that "More" has been selected so that all of the options are shown. In the "Command line" text box, type –lr_citrix_vuser_view.

Performance Testing Citrix Applications Using LoadRunner: Citrix Virtual User Best Practices   6

If the scenario is ran with this option, it will bring up a Citrix window and show the virtual user as it is executing the script. When using this option, do not run a test scenario with many virtual users because the Controller will open one Citrix window for every user in the group. Because this is so resource intensive, it may crash the Controller if used with too many users. Set aside one group with a few users with this setting while the rest of the users are in different groups with standard settings.

Generate Snapshot on Error

The "snapshot on error" feature run-time setting is very helpful when debugging Citrix virtual users. It is found in the miscellaneous options of the Run Time Settings. Anytime a virtual user fails, a screenshot is sent to the Controller which contains an image of what the screen looked like at the point of failure. It is a great way to diagnose syncing problems, or to troubleshoot other strange Citrix errors.
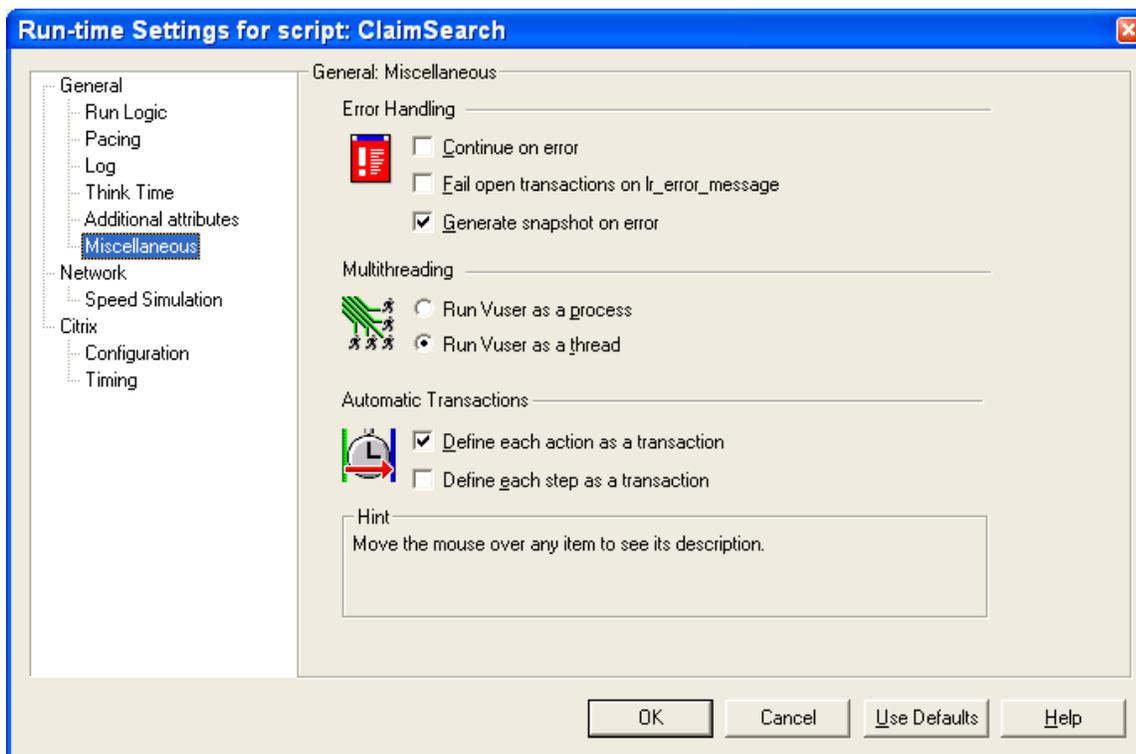
Continue on Error

Feature Pack 2 for LoadRunner 8.1 provides a "Continue on Error" option, which is available for several Citrix functions. It allows individual functions to continue when sync does not work properly. There are several uses for this option. The most important use is for error trapping. If there is a potential for a function failing, the Continue on Error option can be used on the function. In the subsequent lines of the script, the error can be handled appropriately and lines can be written to the log file to assist in debugging.

Scripting without the Citrix Agent

When scripting without the Citrix agent, the script will be composed entirely of bitmap syncs, mouse clicks, and keyboard entry. It treats the application as nothing more than a bitmap. The following are the most commonly used functions.

ctrx_sync_on_window (window_name, event, x, y, length, width, bitmap name)

This function is automatically inserted into the script during recording. LoadRunner recognizes when a main window of the application becomes active or is created. When this happens, it inserts the sync function. The window name argument captures the title of the window being synced. Be sure to check this window title because it may contain hard coded information that could cause the script to break during a later test (e.g. build number, company name). An asterisk can be used in the window name argument as a wildcard. An example of this function is:

> *ctrx_sync_on_window("e-Brokering System *", ACTIVATE, -4, -4, 805, 605,*
> *"87d563f2500432ec6a15b90b2f8003a5", CTRX_LAST);*

> *ctrx_sync_on_bitmap (x, y, length, width, hash)*

This sync function is used when waiting on something other than a window to change. A good example would be a "search" business process. Before continuing with the script, a bitmap sync could be placed in the script to make sure there were results. Inserting a bitmap sync can be done either during or after recording the script. The bitmap sync button is located on the recording bar. To insert one after recording, bring up the tree view of the script. Find a line that has a screenshot of the place that needs a sync. From there, a sync_on_bitmap step can be inserted. The inserted step will look similar to this:

> *ctrx_sync_on_bitmap(449, 90, 107, 18,"9c097ec91de046c32e9b632b16125913");*

The bitmap sync is one of the most difficult parts of a Citrix script to maintain. When they are used, any application changes made in the environment can potentially break the script. If the bitmap syncs stop working, there are a couple of ways to fix the script. The first is to rerecord the sync in a new script and copy the newly captured bitmap sync it into the old script in place of what is there. The second way is to find out what hash value LoadRunner is seeing at the coordinates of the bitmap. Before running the script, turn on advanced logging in the Vugen

run-time settings and enable the advanced trace option. When it gets to the point of failure in the script, the LoadRunner execution log will show the expected hash value and the actual hash value of the bitmap. Copy the actual hash value into the script and determine if the script will run. Alternatively, both bitmaps could be used in the statement by separating them with a pipe ("|"). An example of this would be:

ctrx_sync_on_bitmap(158, 300, 312, 132,
"84e312c9701b3dbb225182ae41cdc228|571b7bccd5eb40172dbc7e79d461528d ");

Feature Pack 2 for LoadRunner 8.1 contains an additional feature to troubleshoot bitmap syncing problems. The "Failed Bitmap Synchronization" dialog box helps to solve the problem of changing bitmaps. If the bitmap values differ from the recorded bitmap values on replay, the dialog box will display. On one side, it displays the bitmap recorded in the original script. On the other side, it displays the bitmap shown during the replay.   The Failed Bitmap Synchronization dialog box provides two options: Stop and Continue. Clicking Stop means that something is wrong with the bitmap synchronization and it will not make any changes to the script. If continue is clicked, the script will continue to replay and the script will be changed to accept both the original bitmap value and the new bitmap value as valid.

ctrx_mouse_click (x, y, mouse button, key modifier, window name)

If the application will not allow a business process to be navigated by the keyboard alone, mouse clicks can not be avoided. During recording, make sure to click slowly through the application and try to click on the center of objects. If clicking on a button, try and click in the middle of that button. This will make the replay more accurate because the mouse clicks are never in exactly the same place. Clicking on menus can be tricky in a script. When clicking on a menu, be sure to allow time in your script for the menu to appear before clicking on the menu option. This can be done by inserting "think time" between the two mouse clicks, or by putting a bitmap sync between the two mouse clicks.

ctrx_type (string)

and

ctrx_key (key, key modifier)

These two functions are the way to send keyboard data to the Citrix server. The ctrx_type function sends normal keyboard data and the ctrx_key function sends non-alphanumeric keys to the server. In order to send special keyboard commands like Alt-F, ctrx_key can be used. For example, ctrx_key ("f", MODIF_ALT) will send Alt-F to the server.

ctrx_get_window_name (string buffer)

Sometimes in an application, different windows appear depending on the option selected in the application. An example of this would be with a "claim search" process in a claim processing application. The script may search for a random claim and then open it. Every once in a while, a claim will not open because of bad data. If this is the case, this data could be gotten by grabbing the window title and then using if/else statements to exit the script gracefully. The following is a code example.

ctrx_get_window_name(window_name);

result = strcmp(window_name, "Bad Claim Data");

if (result == 0) {

   // put code here

}

Scripting with the Citrix Agent

The Citrix agent allows LoadRunner to see the application as more than just bitmaps. After installing the LoadRunner Citrix agent on the Citrix server, LoadRunner can see some of the objects of the application. This allows LoadRunner to perform actions such as syncing on text and syncing on object information. However, it does add about 2 MB to the footprint of a Citrix virtual user – something to consider for the scalability of your LoadRunner Generators. The Citrix agent is only launched when LoadRunner starts the application. If a user starts the application, the Citrix agent will have no effect on his or her session.

Any of the functions used when scripting without the agent can still be used with the agent. Synching with objects provides a more stable script. When recoding with the agent, the agent records mouse clicks differently (ctrx_obj_mouse_click). The only difference is that this function includes a description of the object in its function call.

ctrx_sync_on_obj_info (window name, x, y, attribute, attribute value)

This function syncs on certain attributes of an object before continuing. Some of attributes that it will wait for are enabled, focused, visible, and checked. It can also sync on the text of an object. A text check can be inserted into the script during recording by finding the icon on the recording toolbar. When possible, using a text check would be better then using a bitmap check. A text check would be more easily maintainable.

ctrx_get_object_info (window name, x, y, attribute, buffer for value)

This function is very similar to the ctrx_sync_on_obj_info. It allows the script to capture the attribute of an object. For example, it would allow a script to tell whether a certain checkbox was checked.

ctrx_sync_on_text_ex (x, y, length, width, text, window_name)

This function allows the script to sync on text shown in the screen. It can be added during the recording or after the recording from the tree view. When adding this sync, a rectangle is drawn around the area with the text. During playback, LoadRunner looks for text at the specified location and waits for the text to be shown before continuing the script.

Summary

Because more companies are utilizing Citrix to deploy their legacy client/server applications, performance testing these environments have become more common.   Successfully writing script to imitate a user's action is made fairly easy when using LoadRunner and the Citrix Agent. In order to write a good script, planning must be done to ensure that the proper syncs and the proper checkpoints are in place. Because Citrix requires more synchronization to run properly, the scripts take more time to create than some of the easier protocols like web.

Northway has world class consultants with intimate knowledge of Citrix from an administrator perspective, as well as a performance engineer view. Northway helps companies ensure their applications deployed via Citrix are scalable and meet the performance objectives required for service level agreements.

About Northway Solutions Group

Northway Solutions Group is a technology solution provider that provides consulting services around HP's Business Technology Optimization (BTO) software products. The company is a HP Elite Partner and reseller for HP Software products in the Application Lifecycle Management (ALM) space, which includes Project and Portfolio Management, Quality Center, Performance Center, Application Security Center, and Business Availability Center. Northway employs certified consultants with real-world experience who provide long-term solutions to the toughest business challenges. This includes providing training and implementation services for all the products in our portfolio. Visit www.northwaysolutions.com to find out more information about our services and offerings.

About HP

HP is a technology solutions provider to consumers, businesses and institutions. HP software helps IT organizations make sure that every dollar delivers positive business outcomes. Their business technology optimization (BTO) products automate key processes across IT functions, and their OpenCall platforms enable people to communicate and access services. As the focus of IT shifts from systems management to business results, companies are investing in HP software to lower costs, improve delivery time and reduce risk.

LoadRunner® is a registered trademark of HP Software. All rights reserved. All screen shots used with permission.