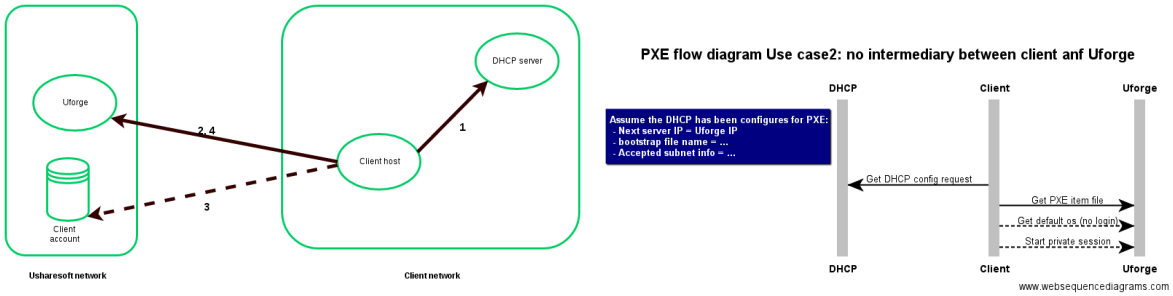
Usharesoft: Internship
PXE environment. Designed architecture

Riyane SID-LAKHDAR

September 11, 2016

Contents



(a) Architecture (b) Network flow diagram

Figure 1: Representation of the solution 2: no intermediary between uforge and client hosts.

1 Software architecture 1: No intermediate between client anf Uforge

1.1 Client Grub interface:

1. **"Secure Uforge choice":** Allows to access the PXE dedicated images on the client's Uforge account. This item requires the user to enter his uforge credentials. This option is automatically selected through a timeout system and requires no user action.

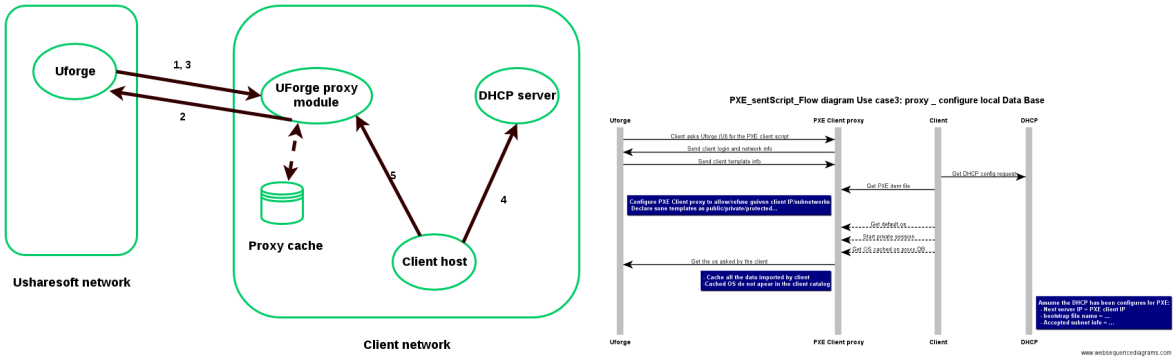
2. **"Uforge default PXE os":** Allows to load the default os image set on Uforge (without any account needed).

3. **"Local cache multi-choice":** Choosing this item allows to access a grub where to chose between different os images. These images must be located on the "PXE server" cache.

1.2 Global architecture

1.3 Advantages

- No need for the client to install or download any new tool (on his network or central server).
- All the improvement and the upgrades may be proposed without any intervention of the user: all the binaries and scripts that uses the user are provided at boot time.



(a) Architecture (b) Network flow diagram

Figure 2: Representation of the solution 3: proxy between Uforge and client hosts.

1.4 Drawbacks

1. Very few features may be added for the client network side.
2. A same file may be sent several times by uforge to the same network hence contention on uforge server. It also downgrades the performances for the client.

2 Software architecture 2: Proxy/Cache between Uforge and client host

2.1 Client Grub interface:

1. **"Private session":** Allows to access the PXE dedicated images on the client's Uforge account. This item requires the user to enter his uforge credentials. This option is automatically selected through a timeout system and requires no user action.
2. **"Uforge default PXE os":** Allows to load the default os image set on the central proxy server.
3. **"Cached multi-choice":** Choosing this item allows to access a grub where to chose between different os images. These images must be located on the "PXE server" cache. They must have been declared by the central proxy as accessible for your IP (or sub network).

2.2 Global architecture

2.3 Advantages

First, this architecture reduces the contention on the uforge server: the the same template may be sent only once for different users (cache on the client Cache). Thus, the client has also a better time performance as most exchanges are done within his local network.

Second, we may allow the user to automatize some installation tasks on his network: he may define the templates that he wants to open. He may also define the groups or sub groups that access to each template.

2.4 Drawbacks

1. Very few features may be added for the client network side.
2. A same file may be sent several times by uforge to the same network hence contention on uforge server. It also downgrades the performances for the client.

3 Implementations

3.1 iPXE web-resources (integrated to UForge)

In order to answer to the pxe request, we have implemented three web-resources integrated within the UForge environment:

- The class "NonAuthentifiedResource.java" allows to get all the PXE binaries and the PXE script that a light user may need before it is able to log to UForge. This web-resource does not requires any authentication. 2 rest URI are linked to this web-resource (see section 4.5.1, and 4.5.2).
- The class "UserImagesResource.java" allows to get the PXE script that prints the prints the interface during a private session (see section 4.6.2). This script is built dynamically using the dynamic code generation tools ().
- The class "ApplianceDownloadResource.java" has been modified to allow to download independent files of an ISO which may have a path of depth higher than 1.

3.2 Dynamic code generation tool (used by the web-resources)

In order to dynamically build the PXE scripts that are sent to the user, we have implemented the tool "ScriptCreator" within the UForge environment (uForgeRest.com.usharesoft.rest.utils).

This use an initial code template that is parsed using a state automaton looking for some key words. Each key word is mapped to one of the actions: write string, loop on an other template, call a function.

3.3 Scripts to set an iPXE environment on an independent server

In order to set an iPXE environment, we have implemented the script "setPxeEnvironment/setPxeServer.sh". The options of this tool may be discovered thanks to the option "-help". This script allows to configure and run different services needed by the PXE specification:

- Configure and launch the DHCP and DNS servers (see section 3.4.2).
- Configure and launch a tftp server (in order to send the initial iPXE binary).
- Configure and launch an HTTP server (to execute all the other file exchanges).
- Populate the http server (or TFTP in case of exclusive PXE) with the sysloader binaries (required by the installer).
- Populate the http server (or TFTP in case of exclusive PXE) with the os files (iso that has been mounted).

Different scripts have been implemented to automatize all this processes. They may all be found in "setPxeEnvironment/util".

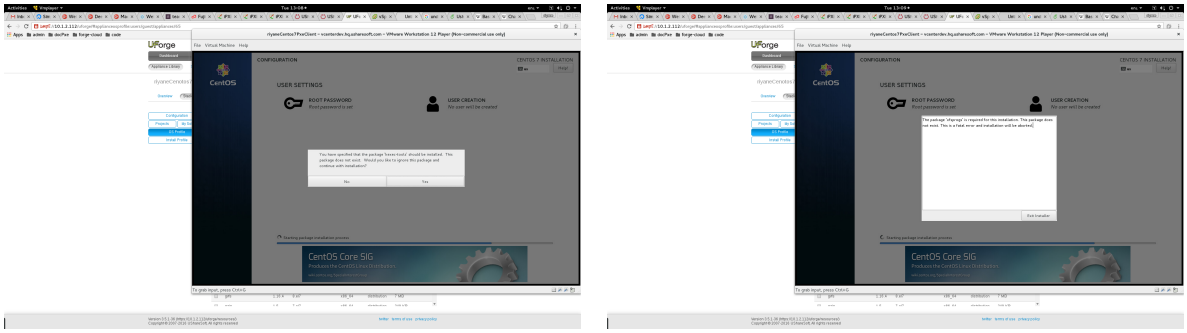
3.4 Example of outputs

3.4.1 iPXE scripts

An example of each script has been put in the directory "examples/pxeScript". This files may be sent as is to the light user in order to print a specific interface. Each one of this script has been dynamically generated by a UForge web-resources as an answer to a specific PXE request. The name of the web-resource java class and the context of the generations are indicated on the head of each file.

3.4.2 DHCP/DNS configuration file

An example of DHCP and DNS configuration files has been put in the directories "examples/dhcpConfig" and "examples/dnsConfig". This configuration files has been respectively used with "dhcpd" and "dnsmasq".



(a) Installer warning

(b) Installer fatal error

Figure 3: Missing packages in the Uforge generated iso

4 General notes

4.1 Issues to boot on a UForge-generated centos7 ISO with a PXE environment

The iso generated on UForge did not boot till the end using the PXE environment. It stopped during the installation (done by the "anaconda" boot loader after the kernel and file system have been downloaded).
Two issues have been noticed during the installation:

- A warning has been shown by the installer indicating the the package ***"kexec-tools"*** has been specified in the profile but does not exist in the iso (see figure 3a). This warning is probably due to a bug during the iso generation (on Uforge) as this package doesn't appear in the required package list (seen on the UForge UI). This warning may be ignored.

- A fatal error has been shown by the installer indicating the the package ***"xf-sprogs"*** is mandatory for the installer to succeed but does not exist in the iso (see figure 3b).

This issues have been fixed by simply adding the previous two packages to the used iso (within the install profile on the Uforge user-interface).

4.2 Issues to boot on a UForge-generated Ubuntu-14.04 ISO with a PXE environment

The problem with this iso is that the path to the kernel (vmlinuz) and the file system (initrd.img) are not the one expected by the boot loader. This issue has been corrected by simply replace them ones the ISO has been mounted at "images/pxeboot/"

4.3 Build the iPXE binary

The ipxe binary that is initially sent to the user has been build based on the open source firmware iPxe (see http://ipxe.org/).
In order to download this firmware, we have implemented the script ipxeFirmware/setIpxeFirmware.sh.
Then, to build the final iPxe binary, we need to into the generated directory "ipxe/ipxe-bios/src" and run the command
" make bin/undionly.kpxe HTTP_AUTH_BASIC=enable DOWNLOAD_PROTO_HTTPS=enable EMBED=<pxeInitialScript> "
where "pxeInitialScript" is the script that you can find at "setpxeEnvironment/config/pxeConf.d/bootstrapScript.ipxe.initial".
Note that a requirement to this build process is to have the packages "zlib-devel" and "binutils-devel".

4.4 General resource (not managed by the web-resource)

4.4.1 Resource iPXE binary

- ***Type:*** TFTP
- ***URL:*** undionly.kpxe
- ***Return:*** The initial iPXE binary.
- ***Details:*** This initial binary (iPXE binary) is supposed to replace the initial PXE binary which was by default burned on the light client network card ROM. It needs to be build according to the process described section 4.3.
  This binary needs to be accessed by all the clients without any authentication. It also needs to be build and configured for the destination client processor architecture.

4.5 REST resource with no authentication (managed by the web-resource)

4.5.1 iPXE script (main menu / private session specific to image)

- ***Type:*** HTTP GET
- ***URL:*** noAuthentication/pxeScriptResource/{sn}
- ***Parameter:***
  - sn: Name of the script to download (may be a path with sevral sub directories)
- ***Return:*** The iPXE script that prints the main menu or the menu for a private session (specific to an image)

- ***Details:*** This script must allow to print the main menu of the iPXE interface. For each item of the menu, it must indicate the correct resource. No authentication is required to access this resource.

4.5.2 Default PXE OS (get independent files of the OS)

- ***Type:*** HTTP GET
- ***URL:*** noAuthentication/pxeDefaultImage/{os-fileName++}
- ***Parameter:***
  - os-fileName: Path to a file of the default os. May be a path with several sub-directories
- ***Return:*** A file belonging to the default os.
- ***Details:*** No authentication is required.

4.6 REST resource with authentication (managed by the web-resource)

4.6.1 User-specific PXE OS (independent files of the OS)

- ***Type:*** HTTP GET
- ***URL:*** /users/{uid}/appliances/{aid}/images/{itid}/downloads/{os-fileName++}
- ***Parameter:***
  - uid: user id (UForge login)
  - aid: Appliance id
  - itid: image id
  - os-fileName: OS file name. May be a path with several sub-directories.
- ***Return:*** The iPXE script that prints the list of os accessible by the user after he is logged.
- ***Details:*** This request is sent after the user has logged him self.

4.6.2 iPXE script for a private session

- ***Type:*** HTTP GET
- ***URL:*** /users/uid/images/scriptPxePrivateSession
- ***Parameter:***

  - Uid: UForge login of the user

- ***Return:*** The iPXE script that prints all the accessible PXE images for the given user.
- ***Details:*** This script must allow to print all the accessible images for the current user. For each item of the menu, it must indicate the correct resource 4.6.1. An authentication is required.