System Installation Suite

Modern Replication

System Installation Suite (SIS) is a collection of tools that allow system administrators to perform automatic Linux system installation. This article describe the usage of SIS from a sysadmin perspective.

BY BENOÎT DES LIGNERIS AND FRANCIS GIRALDEAU

We will examine the interesting functionality for keeping an exact backup of any essential servers (web server, mail server, name server, …) so that this backup can be used to install another server, up and running, in a matter of minutes on any spare box that has similar hardware.

We will expose the basic SIS concepts, then review the necessary steps to make it work in a real environment. Finally, pitfalls and useful advice that will save you a lot of time and frustration will be presented.

Basic SIS architecture

SIS [1,2] is a collection of tools that really looks like a Swiss Army knife for the sysadmin who has a lot of Linux boxes to manage. The principle of operation is simple. First, at least one image is needed. An image is the whole set of files from a Linux system and is stored on the SIS server. Next, you can deploy this image. The auto-installation process can be started on a client with a floppy, a CD-ROM or by a network boot process. The hard disk of the client is then formated and all files from the selected image transfered. Finally, a post-install configuration is done.

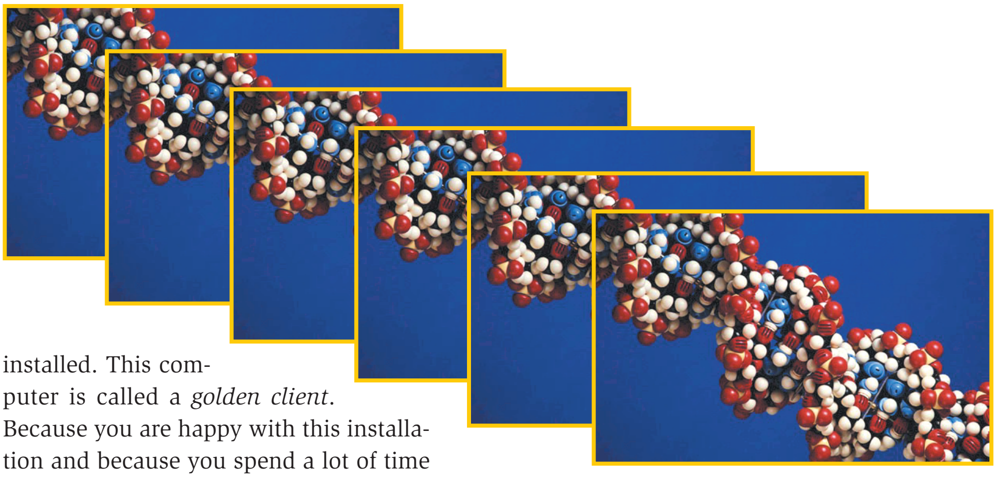
If the image on the server changes, any clients that use this image can be synchronized.

Because it is not a package based method, but a file based one, it is intrinsically distribution agnostic and it can work with any Linux distribution. Images are stored in */var/lib/systemimager/images* and are not compressed. You can modify them before any deployment.

SIS is made of three core components: System Installer [3], System Imager [5] and System Configurator [4] and the functional diagram of SIS is presented in Figure 1.

System Imager

System Imager builds an image from a running computer that you have already installed. This computer is called a *golden client*. Because you are happy with this installation and because you spend a lot of time fine-tuning this system, you want to keep a complete backup of it so that you can, in case of hardware failure of your brand new over clocked 5.6 GHz Linux box, push the image to another similar computer.



This operation is done using the *rsync* protocol and it is not as trivial as it seems! There are a lot of special files and mounted file systems that you don't want to copy. (*/proc*, … or nfs mounts)

System Imager takes care of the auto-installation process by creating auto-install scripts. It offers several tools to create the auto-install media and synchronize the clients with their image on the SIS server (see Image Update later for more details).

rsync is used as a transfer protocol so that you can encrypt the communication with *ssh*. *rsync* preserves network usage, because only files that are new or that have been changed are transfered.

System Installer

This tool provides a convenient way to build an image from a rpm list, without installing a single computer. It is RPM based and, as such is not directly usable on a Debian based system. However, some tools exist for this distribution (*debootstrap* and *chroot*, *rootstrap*, …) that have the same functionality. This is used in the OSCAR [6] clustering project in order to automatically install nodes for a cluster (massive installation).

System Configurator

System Configurator deals with the client post-configuration that occurs at the end of the auto-install process: PCI hardware detection, Network & Network Adapters, Initial Ram-disk generation, Boot-loader, Timezone.

System Configurator manages the configuration files that the distribution uses and makes them functional for the detected hardware and selected options. Officially supported distributions are Red Hat, Mandrake, Connectiva, SuSE, TurboLinux, Debian and distributions that use the same configuration files. The most popular boot-loaders are supported too: Lilo and Grub on IA32 and Elilo on IA64.

Once the System Configurator has been run on the client, at the end of its automatic install, its configuration is done and the node has the exact same functionality as the image. Additional hardware, like USB controllers and soundcard, are not configured by System Configurator: use the tools provided by your distribution to configure this hardware.

This auto-configuration provides a very useful feature because you can install the same image on different hardware. You can have clients with different hard-disks, CPUs, network adapters. They must share the same architecture. When a server is down and you need a replacement as soon as possible, it is a very handy feature.

Basic Image Manipulation

There are many utilities to manipulate images and clients. Some of these tools are available with a Graphical User Interface (GUI) named *tksis*. We will present both aspects of the software as much as possible.

The Command Line Interface (CLI) tools offer more functionality than the GUI and they are completely scriptable. Moreover, *man* pages are well written and you can refer to them for additional information.

Overview

Let's start with the step by step procedure for using SIS:

- Install SIS on client & server
- Prepare the golden client
- Fetch an image from a golden client
- Prepare the auto-install media
- Distribute the image

Note that the steps above are only for RPM based distributions. For Debian, use the *apt-get* tool to install the client and the server. You should be root for all the commands. If the installation fails, don't panic. Watch for missing Perl packages, install the requested packages and then try to reinstall until finally no error messages are produced.

Install SIS on client & server

Go to *http://www.sisuite.org* and click on *install sis now*. Download the script. *chmod +x* it to make it executable and then, run the script as root. Then run *./sis-installer.pl --verbose --download --install client*. The script will download and install the required rpms for the SIS server and/or the golden client. You may have to install other packages to satisfy dependencies. Check the help section of the website for further information about installation.

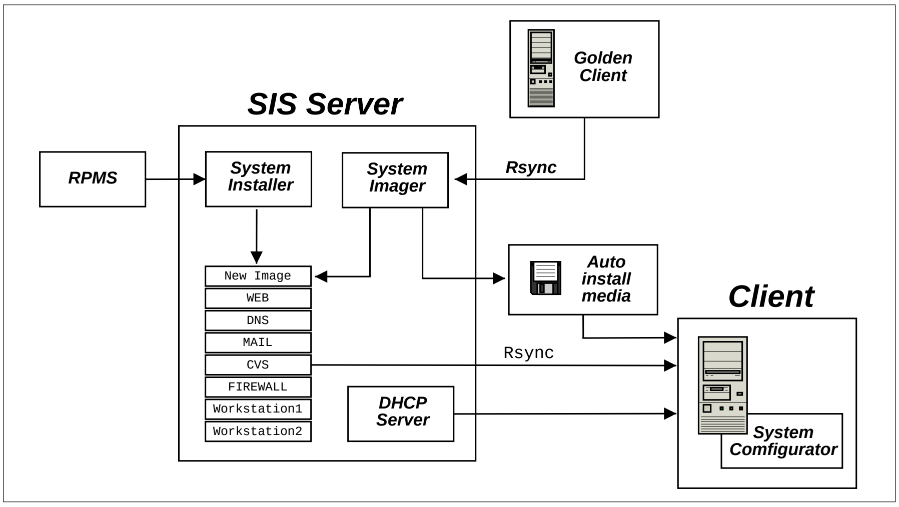


Figure 1: SIS schema

Before installing the server, verify that there is enough free disk space to receive the image of the golden client (*df -h* on the image server and the golden client will tell you if you have enough free space on the image server).

To install the server, follow the above procedure for the client, then run *./sis-installer.pl --verbose --download --install server*. Also, make sure that the *systemimager* service is started:

```
$ /etc/init.d/systemimager start
```

Links necessary for the automatic startup of this service are not made automatically. If you plan to have a dedicated SIS server, please use the tools provided by your distribution, for instance:

```
$ chkconfig --add systemimager➥
 start
```

This procedure is not valid for Debian based system. SIS is available with the *apt-get* tool (package name: systemimager, systeminstaller, systemconfigurator, …). See the section on Security for important advice.

Prepare the golden client

To complete the client preparation, you must run the command *prepareclient*.

```
$ LC_ALL=EN prepareclient
```

You should verify that the execution of *hostname* on the golden-client returns something other than *localhost*. An SSH server should also be installed and running on the golden client if you want to use encrypted communication.

Prepare the SIS server

At this time, you should be able to start the graphical interface with the *tksis* command. The welcome screen will look like the one presented in Figure 2. The fun part starts here.

Fetch an image

It's time to fetch! Click on the *Fetch Image* button and another window similar to the one presented in Figure 3 will pop-up.

For a complete description of the parameters, refer to the documentation of System Imager. In this article we use the *replicant* method whose aim is to replicate exactly the network settings and configuration file of the golden client in order to be able to quickly restore this computer from scratch.

After you have clicked the fetch button, you will be prompted for the password of the user you have selected, but in the terminal window from which you started *tksys*. Be patient, and have some tea.

When the golden-client is fetched, *rsync* builds a list of all the files to be transfered and it can take a while. You can verify that an ssh connection is established between the image server

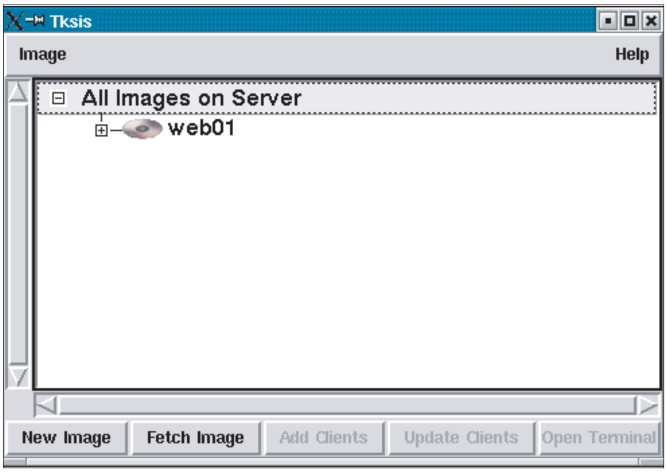


Figure 2: tksys default screen

and the golden client by examining the output of, for instance *netstat -t*.

During the fetching, the *tksis* window is usually not responding: it will come back to a more responsive state once the fetching is complete.

This step can also be achieved using the Command Line Interface (CLI). Here is the CLI equivalent of the fetching operation we performed with the graphical interface.

```
getimage -ssh-user root -image➘
 std-workstation -golden-client➘
 10.0.1.20
```

You can now check the list of available images on the server with the *lsimage* command.

Install the image on another computer

Once there is a SIS image on the server, we can use this image to automatically install any number of clients.

Prepare the auto-install media

You should use the *mkautoinstalldiskette* or *mkautoinstallcd* commands for diskette or CDROM based install media. The default choices should be OK in most cases, otherwise, please consult the man pages of those commands. Alternatively, if you have a *PXE* server, you can output the auto-install media to a file with the *-out-file myImage.img* switch.

Distribute the image

This is the simple part of the process: simply put the auto-install media in the computer you want to install and make certain that the auto-install media is the first booting device in the BIOS of the client.

The install process should start and the computer will be completely formatted and installed with the exact same files as the image on the SIS server. You can watch the progress of the installation on the console if the computer has a screen.

Pitfalls, useful advices and advanced usage

Here are some direct experiences of SIS usage: some are very positive ones whereas others were very time consuming and frustrating.

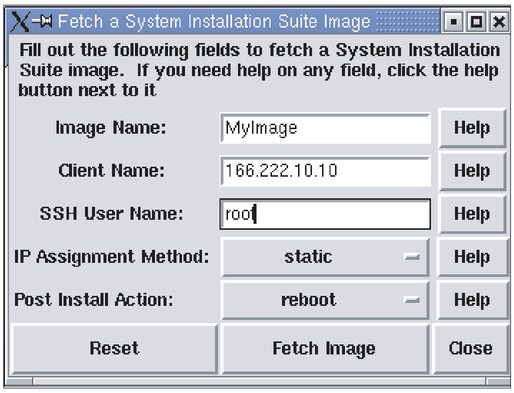
Security

The *rsync* daemon on the server exports all images to all computers anonymously: all the image files are freely readable by everyone. It is possible to restrict access to these files. See *man rsyncd.conf* for details, */etc/hosts.allow* and */etc/hosts.deny* if you use *tcpwrappers* or your firewall rules to restrict access. Also, see the *systemimager-ssh* package to make the SIS server secure.

Image update

There are several ways to update an image. The first one is to follow exactly the same steps carried out in the Prepare the SIS server section. SIS will detect that an image with the same name already exists and ask you if you want to update this image. If you say yes, the image will be updated and because *rsync* is designed for this kind of problem, the update will be very quick as only the files that have changed will be transferred.

The other solution is to alter directly the files in the directory where the image is stored (such as. on the image server). While this is not advised, you can do it at your own risk. In this case, the *chroot* command will save you a lot of trouble. Don't forget to test your changes before applying them in order to ensure that your image is fully functional.



Fill out the following fields to fetch a System Installation Suite image. If you need help on any field, click the help button next to it

Image Name: MyImage
Client Name: 166.222.10.10
SSH User Name: root
IP Assignment Method: static
Post Install Action: reboot

Figure 3: Fetching an image

Client Update

Now that you have updated your image, you are ready to distribute those changes to a computer that has the older image. In order to do so, you will use the *updateclient* tool.
This tool is provided in the client programs of the SIS tools, this should be installed in your image! You can even automatically update your clients by adding an entry to cron:

```
updateclient -a -s imageserver➘
 -c eth0
```

With this system, clients will be updated periodically without the need for any human interaction. A push mechanism exists, but in that case, you have to use another replication method than the one we have used in this article (such as.: dhcp-static, dhcp-dynamic or static) and then use the *addclient* procedure. Then you will be able to use the *pushupdate* command on your server.

Another DHCP server on the same subnet ?

The client being automatically installed assumes that the image server is the DHCP server unless specified otherwise. It is necessary to overwrite this setting if you want to keep a separate DHCP server running in your network. In order to customize this setting, add a custom *local.cfg* file into the root of the boot disk (otherwise, the client

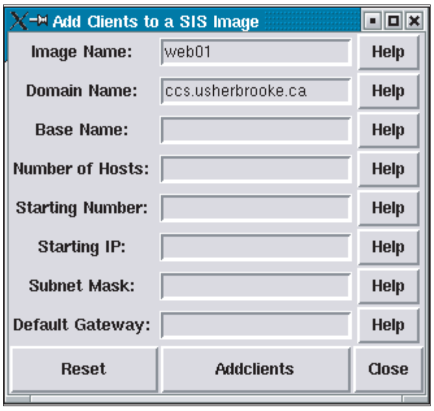


Image Name: web01

Domain Name: ccs.usherbrooke.ca

Base Name:

Number of Hosts:

Starting Number:

Starting IP:

Subnet Mask:

Default Gateway:

Figure 4: Add clients (*replication* of images)

will try to contact the *rsync* service of the DHCP server, which will certainly be closed unless you used *rsync* on your DHCP server). Here is an example of this file:

```
# FILE local.cfg
IMAGESERVER=10.0.1.3
IMAGENAME=workstation
```

Replace the IP by the IP or name of your image server and the imagename by the name of the image you want to install.

Alternatively, if you want/can modify the configuration of your DHCP server, you can use the *option-100* parameter and set its value to the IP address of the image server.

From a dual Xeon to a mono-Pentium III

One of our goal, using SIS, was to transfer our nameserver (DNS) from a dual Xeon computer to a mono-Pentium III computer. While this is not officially supported *out of the box* by SIS, the migration was quite easy.

INFO

[1] System Installation Suite web site: *http://www.sisuite.org/*

[2] System Installation Suite, Massive Installation for Linux: *http://sisuite.org/presentations/OLS_2002/sispaper.ps*

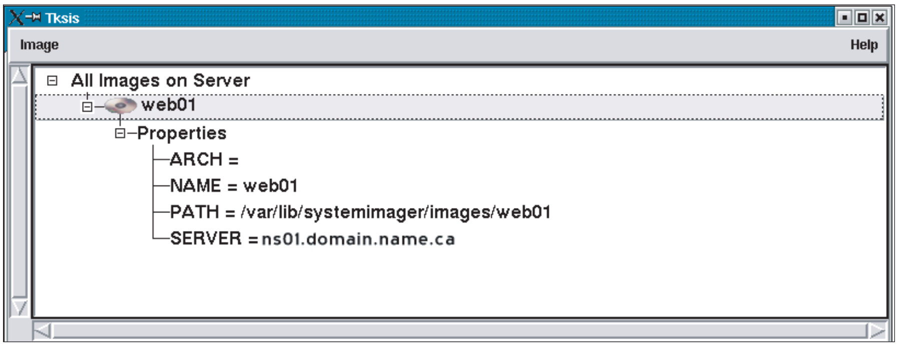
[3] System Installer web site: *http://systeminstaller.sourceforge.net/*

[4] System Configurator web site: *http://sisuite.org/systemconfig/*

[5] System Imager web site: *http://systemimager.org/*

[6] Open Source Cluster Application Resources (OSCAR): *http://oscar.sf.net/*

[7] Centre de Calcul Scientifique, Université de Sherbrooke: *http://www.ccs.usherbrooke.ca/?lang=EN*



```
⊟ All Images on Server
   ⊟─ web01
      ⊟─Properties
          ├─ARCH =
          ├─NAME = web01
          ├─PATH = /var/lib/systemimager/images/web01
          └─SERVER = ns01.domain.name.ca
```

Figure 5: Selected image – what can be done

SIS will not detect CPUs and will not try to adapt the kernel to the material in use: it assumes that the architecture and the number of CPUs are the same. However, because it is possible to install more than one kernel (and choose it with your boot-loader), a mono processor kernel was installed on the dual-CPU golden-client which was then fetched.

Then we install our mono-PIII without problem. The reboot failed with the SMP kernel. However, on the next reboot, we were able to choose the mono-CPU kernel and all went well. After the first boot, we selected this kernel as the boot-loader default choice.

Minor hardware changes

All the hardware not managed by SIS was not functional: USB controller, sound controller, video card. In order to have them working, we used the automatic detection tools provided by the distribution: all those elements were success- fully configured before the end of the first boot. Make sure you have the hardware detection program installed and running in your SIS image.

Modification of the partition scheme and disk migration

With SIS, you can specify the partition scheme before installing clients. This is a very versatile feature, that enables the administrator to adapt partitions for his particular needs very quickly.

The file that contain information about the disk partioning is in the file *etc/systemimager/autoinstallscript.conf* in the image directory */var/lib/systemimager/images/myimage*. This file uses XML markup and is easy to understand. For example, you can modify the size or the file system of a partition.

Another great example of what can be done with SIS is changing the filesystem type of a server. It is always hazardous and often painful, but with SIS, you can do it in a fully automated and reproducible way. Simply modify the *autoinstallscript.conf* by changing the *fs* attribute of the *fsinfo* tag.

Conclusion

We have seen that SIS is a very powerful tool with no real equivalent in the Linux world. The ability to completely clone a computer and then deploy this image on another similar box is a very interesting capacity. In the case of a hardware failure (such as the hard disk, motherboard, …) you can rebuild the exact same server with its functionality on another box in a few minutes.

This suite of software is also very interesting for OEMs (automatic installation of image) and is a core component of the OSCAR clustering solution in order to build an image (from a list of RPMs) and massively deploy it on cluster nodes. ■

THE AUTHORS

Benoît des Ligneris has a Ph. D. in physics and is a Linux cluster-builder. He works part-time for the Centre de Calcul Scientifique of Sherbrooke University, Canada [7]. Benoît is also core member of the OSCAR clustering project, chair of the thin-OSCAR workgroup and co-creator of the LAMP project MyDynaWeb.

Francis Giraldeau is a student in Electrical Engineering at the Université de Sherbrooke. He works on thin-OSCAR, the tool to make diskless clusters within the OSCAR framework.