

Automatic Installers Review

Emir Imamagić

Davor Mihajlović

*Department of Computer Systems,
University Computing Centre, Croatia
{eimamagi, miha}@srce.hr*

Abstract

Automatic Installer is a system used for installation and configuration of OS and various software packages on large number of remote computers. We divide such systems in two groups: image based and package based.

In this paper, we describe automatic installer system in details. We explain differences between two groups and give overview of existing Linux based automatic installers. Furthermore, we emphasize importance of such systems in process of computer cluster installation. In addition, we describe our own experiences with some automatic installer systems.

1. Introduction

Cluster computer consists of a group of interconnected computers working together as a single computer. Individual computers that form the cluster are called nodes. Usually all cluster nodes have the same software stack installed and the same hardware configuration. Because of that, installation of nodes is performed using special systems for remote installation. These systems are called automatic installers.

Automatic installer is a system used for automatic deployment of software stack on remote computers. Automatic installer system consists of one server computer and

numerous client computers. Server computer contains software stack and additional programs needed for remote installation of software stack. Client is the computer on which the software stack is installed.

Beside for cluster installation, automatic installers are used for installation of large number of workstations. For example, automatic installers are used for installation and configuration of classroom computers, Internet server farms and large number of workstations at institutes or corporations. When used in such manner, automatic installer has to install various software and OS stack on various platforms. In case of clusters, installation is much simpler. Cluster nodes usually have similar architecture and software stack is the same. Consequently, automatic installer used for cluster installation does not have to have advanced features.

In this paper, we examine different automatic installer solutions from the perspective of cluster nodes installation. We are limited to Linux based automatic installers.

Rest of the paper is organized in following way. Second section describes automatic installers in details and our classification of automatic installers. Third section gives overview of set of existing systems based on online references and

manuals. In fourth section we shortly describe our own experiences with some of existing systems. In fifth section, we summarize our experiences and emphasize automatic installer solutions we find appropriate for cluster installation.

2. Automatic Installers

Automatic installer is a system that enables automatic installation and configuration of OS and software packages on remote computers. In the automatic installer system, we distinguish two types of computers: server and client. Server is the computer on which we define and hold software stack to be installed. Client is the computer where the software stack is installed.

Automatic installer has following architecture. Server computer has tools for defining software stack that is supposed to be installed on client computers. Client installation is usually performed in following way. Client uses PXE and TFTP protocols to get certain kernel image. After initial boot, client uses NFS or some other network protocol to get OS configuration and software stack.

We divide automatic installers in two groups: **package** based and **image** based. These two groups differ in the way they store software stack on server computer. Package based automatic installers hold software stack in some package form. Examples are RPM and Debian packages. User defines list of packages that are installed on target client systems. Image based installers hold software stack in form of complete system image. User can mount client image on server and test or modify software or configuration.

2.1. Package vs. image based automatic installers

Major advantage of package based automatic installers is easy update of clients and easy modification of installed software packages. If user needs to install new client with small differences in software stack, he just has to define additional packages for that client. Major problem is lack of common packaging method that would make package based installers OS independent. In addition, it is hard to make small configuration changes in software packages because user has to write new packages for that, or modify post-install scripts for individual packages. Another problem is dependencies between individual packages. Package based automatic installer has to provide mechanisms for checking if all dependencies in list of packages are satisfied.

Image based approach is much more intuitive than package based in a sense that the whole client image is provided. User can mount client image from server and test it or make changes in configuration. In addition, since the whole image is stored on server computer, image based installers are OS independent. Disadvantage is installation of different software suites on clients, because user has to make a completely new image for every software suite.

Client infrastructure determines the choice of automatic installer type. In case of heterogeneous client infrastructure, package based automatic installer would be more appropriate. With image based automatic installer, storage space is proportional to the number of different client software stacks.

3. Overview of automatic installer solutions

In order to choose optimum automatic installer system for cluster installation we have evaluated set of systems. All evaluated systems are used for installation of different flavors of Linux OS. Installers here examined are LUI, SystemImager, System Installation Suite, RedHat Kickstart, LCFG and FAI.

As far as we know, there are similar systems for many other platforms. Some of them are Remote Installation System for Windows [8] and JumpStart for Solaris.

3.1. LUI

Linux Utility for Cluster Installation (LUI) [3] was a project started by Rich Ferri, contributed by IBM Linux Technology Center. Project ended in 2001.

LUI is a package based automatic installer. In LUI, everything is a resource: RPM packages, disk partition tables, post install scripts, kernels, client computers etc. Resources are abstract descriptions placed in LUI database. Client software stack is defined by assigning set of resources to the nodes. It is also possible to assign resources to group of clients.

Installation process is following. Client is booted via PXE protocol. Once booted, client mounts NFS root on LUI server. After that, installation script uses the information from LUI database to finish the installation.

Major issues of LUI are unreliability and complexity of resource model. Dependencies between packages are not checked so installation would stop in case of unsatisfied dependencies. Definition of client software stack demanded from user to understand and defined complex set of abstract resources.

3.2. SystemImager

SystemImager [5] is a project started by Brian Elliot Finley. SystemImager is an image based installation tool. SystemImager system is primarily used for installation of homogenous set of computers.

On SystemImager server, image of system is held. Image of system is created using special node, called *golden client*. *Golden client* is template for all other clients. Client installation process is following. Client uses network boot process (PXE) to fetch special Linux kernel – BOEL (Brian's Own Embedded Linux). After loading, the BOEL kernel fetches auto-install script from server. Script configures hardware options (e.g. disk partitions) and then uses rsync tool to fetch and copy the image from server.

Major issue of SystemImager is inflexibility. Simple difference between client nodes requires building a new system image. Another problem is that *golden client* has to be installed manually for each different hardware configuration.

3.3. System Installation Suite

In the spring of 2001, LUI was merged with SystemImager into System Installation Suite[6][4]. System Installation Suite solves issues of both systems that are mentioned above. System Installation Suite is mix of package and image based. Software stack is placed in form of image on server, but the image is created using packages.

System Installation Suite consists out of following three subsystems: SystemInstaller (former LUI), SystemImager and System Configurator. SystemInstaller uses much simpler resource model to build client images from a set of RPM packages. System Configurator is addition to SystemImager

and it is responsible for hard disk, network, boot and ramdisk configuration on client.

Major System Installation Suite weakness is RPM package dependency. However, SystemImager part can still fetch image from *golden client*. That was user could still create images by circumventing the SystemInstaller component.

OSCAR cluster distribution uses System Installation Suite for installation of cluster nodes.

3.4. FAI

Fully Automatic Installer (FAI) [1] is an automatic installation tool for Linux Debian OS. FAI is package based system that uses Debian packages.

Configuration of clients is held in configuration files on FAI server. Client installation procedure is similar to LUI's. Client is booted using network boot. Once booted, client mounts root file system via NFS from FAI server. After that, install script uses Debian apt-get tool to install software packages.

Major disadvantage of FAI is that it can be used only with Debian OS.

3.5. RedHat Kickstart

RedHat Kickstart [7] is RedHat mechanism that enables users to define all parameters needed for installation of RedHat OS. Kickstart is primarily used for automation of installation on local computer, but it can be used for remote installations. Configuration is made in specially defined kickstart files. Client computer accesses kickstart file by using network filesystem, http or tftp. PXE can be used for further automation of install process.

Major disadvantage of RedHat Kickstart is that it can be used only with

RedHat OS, being an integral part of the RedHat distribution.

RedHat Kickstart is used for installation of cluster nodes in NPACI Rocks cluster distribution.

3.6. LCFG

Local Configuration System (LCFG) [2] is a project started by Paul Anderson at Edinborough University. LCFG is package based automatic installer. Beside automatic installation, LCFG can dynamically configure and update remote machines.

Software packages are kept in form of RPM packages on LCFG server. Clients are described with special language in *source* files, which are held on LCFG server. Web interface is provided on LCFG server and it is used to display status of installed clients. Client installation process is similar to FAI. Once the client is up, LCFG Client service is started. Every time when configuration of node on server is changed, Server service notifies all necessary Client services. Client services then update the machine.

Major issue of LCFG is complexity of source files. When defining client software stack, user has to learn source files language. Advantages of LCFG are dynamic update of software and configuration, web portal interface and reusability of source files. It is possible to create more *source* files for specific parts of system and use them to describe clients. That way user can easily describe set of clients with similar software stack.

LCFG is used as deployment tool in European DataGrid (EDG) project [9].

4. Experiences

In various projects, we have experimented with following automatic installers: FAI, SystemImager, System

Installation Suite, RedHat Kickstart and LCFG. SystemImager was used purely for testing purposes. RedHat Kickstart and System Installation Suite were used for cluster installation as a part of OSCAR and NPACI Rocks distributions. LCFG was used for installation of EDG services on Srce institute.

4.1. SystemImager

FAI and SystemImager were chosen for testing purposes because they support Debian Linux distribution. The idea of creating Debian Beowulf cluster distribution became attractive because of its simpler package management system and upgradeability, which simplifies system administration and maintenance of Linux Cluster. Besides that, these automatic install systems could be used to speed up disaster recovery process by shortening the base OS installation time on a new computer. By the time initial problems of installation were solved, a question of distribution maintenance arose, because Srce lacked human resources that would support such project. SystemImager has proven to be easier to install, but lacks the configuration possibilities that FAI offers.

4.2. RedHat Kickstart

RedHat Kickstart is a part of NPACI Rocks [10] cluster distribution, which has been chosen for installation of Isabella Beowulf cluster on Srce [12]. Isabella cluster still runs on NPACI Rocks and Croatian research institutes use it for their HPC needs. Srce still uses NPACI Rocks for installation of its test and production clusters.

4.3. System Installation Suite

System Installation Suite was used as a component of OSCAR [11] cluster distribution for installation of test cluster.

OSCAR was chosen as one of the test cluster distributions. OSCAR was used for installation of cluster with five nodes. Furthermore, mounting the node image on server was useful for installation of applications that are not supplied in form of packages. Mounting the node image was also useful for making small changes in configuration.

4.5. LCFG

We used LCFG system for installation of EDG services on Srce computers. In the beginning, we investigated the possibility of installation LCFG clients on top of existing OS's. In literature, that installation is called LCFG Lite. From the beginning, we had many problems: lack of EDG's support, RPM conflicts on clients and general lack of documentation for that installation type. In the end, we decided to use computers without OS.

Once we decided to follow the EDG documentation, installation procedure was relatively easy. After EDG services installation, we experimented with adding new software packages and dynamic update and configuration.

5. Conclusion

Based on information we gathered, our conclusion is that following three systems have satisfying properties to be used for automatic installation: System Installation Suite, FAI and LCFG. System Installation Suite can be used for installation using various types of RPM packages (e.g. RedHat, Mandrake). FAI can be used for deployment of Debian packages. LCFG is chosen because of dynamic update and configuration features.

Our opinion is that image based automatic installers are generally better for cluster nodes installation. Cluster nodes are mainly homogenous and the software stack

installed on them is completely the same. Image based automatic installers have the advantage for such installations because user does not have to learn special languages for software stack definition. User can just create one image and then distribute it over the nodes. In addition, our experience shows that the most of the cluster tools are not supplied in package form. Using image based tools, user avoid package creation and simply install the necessary tools on *golden client* and creates the image on server, which is then distributed on the rest of the nodes.

6. References

- [1] Fully Automatic Installer, <http://www.informatik.uni-koeln.de/fai>
- [2] LCFG, <http://www.lcfg.org>
- [3] LUI, <http://www-124.ibm.com/developerworks/projects/loi/>
- [4] S. Dague: "**System Installation Suite Massive Installation for Linux**"
- [5] SystemImager, <http://www.systemimager.org>
- [6] System Installation Suite, <http://www.sisuite.org>
- [7] "**Kickstart Installations**", The Official Red Hat Linux Customization Guide, pp. 5-38
- [8] RIS Overview and Tools Requirements, <http://www.microsoft.com/whdc/archive/risover.msp>
- [9] The EuroGrid Project, <http://www.eu-datagrid.org/>
- [10] NPACI Rocks <http://www.rocksclusters.org/>
- [11] OSCAR, <http://oscar.openclustergroup.org/>
- [12] Isabella Beowulf cluster, <http://www.srce.hr/isabella/>