A MODULAR ADMINISTRATION TOOL FOR LINUX COMPUTERS

D.R. Yocum, C. Sieh, and D. Skow
Fermi National Accelerator Laboratory, Batavia, IL 60510
K. Bauer
Georgia Institute of Technology, Atlanta, Georgia 30332

January 13, 2000

Abstract

Managing large clusters of computers (i.e., 500+ nodes) is a difficult task to accomplish by hand and currently there is no single coherent set of tools in the commercial, government, or open source sectors to assist system administrators in such a daunting task. Various efforts at Fermilab are working to create a framework for a set of software utilities to administer these large Beowulf-class clusters of computers. We focused this project primarily on developing an utility to manage the configuration parameters and packaged-installed software. Due to our design criteria, such a utility will also be useful in managing both individual and clusters of systems.

Testing was performed on existing technologies to determine their potential usefulness and the decision was made to combine several products with new processes and programs utilizing a modular design. Development focused on designing and implementing the core module for an individual system, with remote, centralized repository software targeted as a future phase of development.

1 Introduction

Tracking configuration and software package changes on an individual system is difficult. Tracking all the changes in a large (500+ node) cluster is Herculean. Add to that the fact that most institutions have multiple system administrators, differing hardware configurations, multiple distributions and different distribution versions, as well as custom software installations and the problem becomes nearly impossible. We have determined that a utility must be created that can automatically tracking these changes and restore them when problems arise, including catastrophic system disk failure. A system administrator would then be able to recover a system by installing the baseline software (i.e., Fermi Linux

x.x.x) and restoring the configuration and software changes that the utility had previously recorded.

Our intent has been to create a framework for such a utility that can automatically record, track and restore changes to an individual system's configuration files and installed set of software packages. We designed the utility to be modular to accommodate several software package formats. Currently the pilot software package tracking module is for RPM packages, though creating modules for tracking Debian, Solaris, and IRIX packages should be relatively easy.

2 Goals

The utility we have developed, SysTracker, has been designed to function adequately on a standalone machine by storing changes to configuration files and software packages to a local disk at arbitrary intervals, such as every night. These "saved states" are then defined to be singular checkpoints. The checkpoints can then be archived remotely to a central repository server for redundancy. As such, SysTracker has been designed to be scalable from one machine to thousands, given that there are several central repository servers in the latter case.

Our design dictates that SysTracker should use as few system resources as possible, particularly disk space, on both the local machine as well as the central repository. To accomplish this, changes to text configuration file are stored in RCS repositories and only a single copy of any software package type and version is archived.

We have also allowed for flexibility of administration techniques. This allows for the use of any administration tool to be used to manage and maintain a cluster of machines, including manually editing files on individual systems and installing specialized software packages. Any change that has been made will then be recorded when Systracker is run.

Any machine using SysTracker will have the ability to restore to any previous checkpoint, not just the last known good state. The use of RCS facilitates this requirement.

Setup and maintenance of SysTracker should be as automatic as possible such that a non-experienced system administrator can install and run it with the "push of a button." As a further requirement, if SysTracker is used in a cluster environment the system administrator should not have to inform the central repository server of the new node.

3 Research

In order not to replicate the efforts of others, several existing products were investigated as possible basis in the utility. The following is a list of those products with a brief summary as they pertain to the problem.

BOOTP BOOTP is a IP/UDP bootstrap protocol which allows a client machine to discover its own IP address, the address of server host, and the name of a file to be loaded into memory and executed (Croft 1985). Currently, the use of this protocol is beyond the scope of our project but should provide to be useful if it is determined that a machine should have the ability to reboot and rebuild itself automatically via the network.

RPM RPM is the Red Hat Package Manager which is a tool for managing multiple software packages on a Unix-like system. It is primarily used on Linux systems, but has been ported to other Unices, including AIX, Solaris, and IRIX (Bailey 1997). Since we are primarily interested in managing Linux machines, and specifically chosen the Red Hat distribution as our base for Fermi Linux, this is the first package management system we are utilizing.

RCS RCS (Revision Control System) has been in use in code development for many years to archive changes in source and lends itself easily to maintaining changes to text configuration files. SysTracker relies heavily on this product.

cfengine cfengine is an abstract programming language for system administrators used on huge, heterogeneous networks (Burgess 1998). It provides an easy and elegant way to maintain complicated networks and may be a solution to help merge other tools into SysTracker, however, we are currently not implementing it in our design.

prsh prsh is a parallel asynchronous interface to rsh or ssh. It runs a command on a set of remote machines in parallel, without blocking (Salmon 1999). This holds great promise to be included in the final SysTracker product.

cfm cfm is a configuration file manager utility (Metzger 1999). The claims of this product are interesting, but currently our assessment is that this is a work-in-progress and not currently usable.

rdist/rsync Rsync and rdist are essentially the same tool, except that rsync is more efficient. Rsync uses a quick and reliable algorithm to very quickly sync remote and host files since it only sends the file differences of the files across the network rather than an entire file (Tridgell 1999). This product should be very useful in the final remote archival procedures of SysTracker, and it will be incorporated at the time that it is needed.

synctree Synctree is a utility to keep a set of machines installed with a consistent set of software and configuration parameters (Larke 1998), however, since it is dependent on AFS, we have not considered integrating it with SysTracker.

AutoRPM AutoRPM is a program that can keep installed RPMs consistent with and FTP site or local directory and keep installed RPMs in a cluster or network of systems consistent (Bauer 1999). AutoRPM is already heavily integrated into the Linux deployment at Fermilab to distribute software patches to the Linux systems on site and it will certainly fit into SysTracker as a module to distribute software upgrades.

4 Design

The design of the utility consists of several programs that determine the differences in configuration files, programs, directories, data files and logs when compared to archives of those items which were previously stored. The system administrator determines what items should be archived by specifying those items in a configuration file that SysTracker parses.

SysTracker tracks changes that occur to files that belong to software packages as changes against the individual file rather than against the package that contains the file, thus reducing the amount of data that needs to be archived. It also utilizes RCS to archive changes to text files such as those found in the /etc directory: fstab, inetd.conf, conf.modules (Linux only), exports, etc. By default, RCS stores changes to files incrementally which allows an administrator to obtain differences from any previously archived checkpoint. This allows for great flexibility in a system restoration procedure. The same is true for installed sets of software: an archived database of what software packages were previously installed as well as an archive of the packages themselves will save copious amounts of time during restoration.

To restore a machine, the administrator would only have to re-install the baseline distribution (i.e., Fermi Linux x.x.x) and restore the changes from a specific checkpoint stored in the archives.

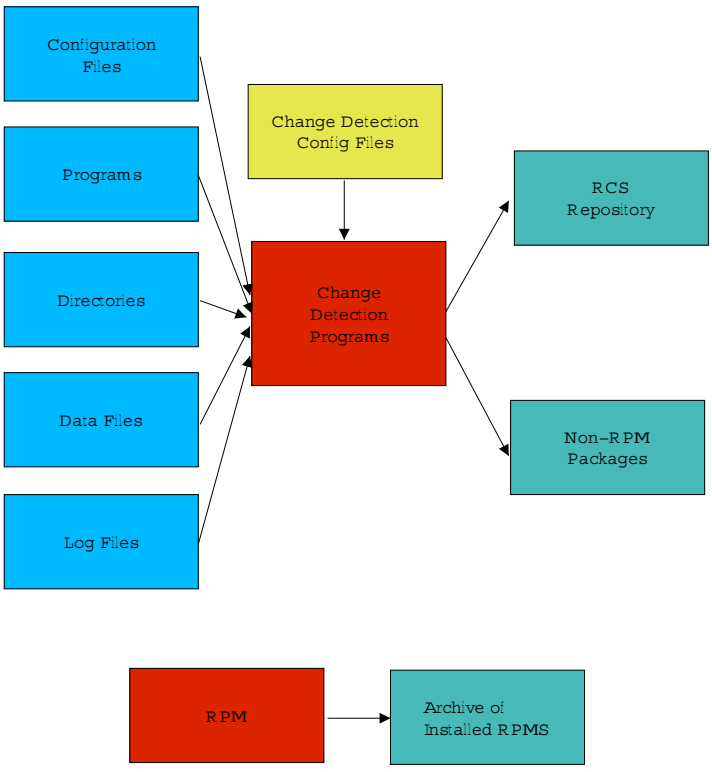
We have modified RPM to automatically store software packages as they are installed on a machine, thus creating a local repository of the installed software packages. This local archive can then be mirrored on a central, remote repository server. The individual system database of the installed software packages can also be stored on the central server; a separate database for each system.

As stated previously, SysTracker has been designed to be modular to accept programs that provide new functionality when they become necessary. We are then able to design the core programs that run on the individual (local) machines and the programs that run on the central archive server separately. No assumptions about the machines have been made regarding OS type or version, which allows for the handling of many package format types such as RPMs, inst, pkgadd, dpkg, tar archives, etc.

Figure 1 outlines the method of operation of SysTracker. The boxes on the left represent the items that need to be archived for disaster recovery. The center portion is the locally installed SysTracker which monitors and records changes to those items which are specified in the SysTracker configuration file.

The right boxes are the local repositories of the configuration file changes and the installed software which has not been installed via a package management tool such as RPM, inst, or pkgadd. The lowest portion of the diagram represents the archive of the software that has been installed with a package management tool.

System Architecture



5 Development

Perl was chosen for development due to its high portability. Development focused on the core individual (local) machine functions.

The components of SysTracker are as follows:

SysTracker This is the main program. Its functions are to parse its configuration file, maintain log file support for concurrency and restart capabilities, and support plug-ins for modularity. All other plug-ins must use this program.

ConfigTrack This plug-in handles configuration file changes. It takes the information from the configuration file parsed by SysTracker and monitors and records changes to the files specified.

StandardTrack This plug-in handles file changes. It takes the list of files and directories from the configuration file parsed by SysTracker, archives their permissions and ownerships and logs those changes.

RPMTrack This plug-in monitors installations, upgrades and deletions of RPM packages. It maintains a database of the package names and the time they were modified.

MD5Track This plug-in maintains an MD5 database of files that have been changed.

6 Future Development

Further development will focus on the design and development of software to automatically archive the local differences database on a remote server which will help facilitate system restoration in the instance of catastrophic system disk failure. The system administrator will be able to install the baseline installation and then, using the archived packages and configuration files, restore all changes from the remote archive server.

7 Conclusion

The process of creating a framework for a suite of utilities to manage large clusters of computers easily and efficiently is underway and proceeding on fronts addressing system installation, monitoring, and disk and file management. We have developed a pilot utility to store changes to configuration file and track installed software packages. SysTracker has been designed to be modular to accommodate many different package installation formats (RPM, inst, pkgadd, etc.) as well as hereto unforeknown features. It has been designed to archive, via RCS, changes to text configuration files, log files, directory structures and other manually installed software allowing for a wide choice of tools for installation and configuration.

References

[1] Bailey, E.C. 1997, Maximum RPM, 3

[2] Bauer, K. 1999, AutoRPM, http://www.kaybee.org/~kirk/html/linux.html

[3] Burgess, M. 1998, Automated system administration with feedback regulation, http://www.iu.hioslo.no/ mark/research/feedback/

[4] Croft, B., & Gilmore J. 1985, RFC 951

[5] Larke, J., & Lockard, J. 1998, Synctree, ftp://ftp.math.lsa.umich.edu/pub/Synctree/

[6] Metzger, D. 1999, CFM, http://www.remotesensing.org/cgi-bin/cvsweb.cgi/sadm/

[7] Salmon, J. 1999, Prsh, http://www.cacr.caltech.edu/projects/beowulf/Grendel-Web/software/index.html

[8] Tridgell, A., & Mackerras, P. 1999, Rsync, http://rsync.samba.org/