SUPPORT OF SNAPSHOT AND ROLLBACK OF CONFIGURATION FILES IN FEDORA

Jan Včelák
Bachelor Degree Programme (3), FIT BUT
E-mail: xvcela03@stud.fit.vutbr.cz

Supervised by: Aleš Smrčka
E-mail: smrcka@fit.vutbr.cz

Abstract: The paper introduces a new tool for versioning of configuration files of a GNU/Linux system, namely focused on Fedora distribution. It is designed to eliminate problems of existing solutions while keeping their benefits. In addition, it brings some new and useful features like synchronization with remote server and integration with other management and configuration infrastructure tools.

Keywords: configuration, versioning, Fedora, Sandglass

1 INTRODUCTION

System configuration files on Unix operating systems are usualy stored as textual files in directories according to *Filesystem Hierarchy Standard* [1]. In many cases, this means `/etc` directory. Creating backups of these files is essential from several reasons, one of which is a possibility of restoring previous versions and comparing several versions of configuration files. All of these operations are fully in the hands of a system administrator.

2 PRESENT SITUATION

The simplest solution is to create a backup of a configuration file immediately before the administrator starts with modifying it. Copying the file to the same directory and appending some suffix is standard practice. After the modification is complete and verified, the administrator deletes the backup file or leaves that backup to be overwritten next time.

Creating and handling own backup files manually is uncomfortable and the administrator has to be careful not to modify files before backing them up. This will become more difficult when multiple files need to hold some change. This approach makes another claims to administrator's concentration and disturbs from the primary goal.

Since configuration files ale textual files, other and more sophisticated solution is provided via version control systems (initially designed for source code versioning). The administrator can easily create a repository from the directory with configuration files. Such a solution is also provided by several tools, namely *etckeeper* [2] which uses versioning backends like *Bazaar*, *Git*, or *Mercurial*. It also keeps other metadata, which are version control systems unable to track, like files permissions and owners. It also utilizes package manager hooks to react to certain operations, e.g. package installation.

Usage of version control systems brings a lot of advantages. Creating a revision will store the state of all files (snapshot) and the administrator can easily modify as many files as needed. In addition, browsing and inspecting the changes can be easily performed via tools provided by version control system. Tool *etckeeper* increases the usability of this solution by bringing more automation. On the other hand, version control system will pollute configuration directory with metadata files—this is mostly undesirable since it violates the *Filesystem Hierarchy Standard*. And system administrator has to be aware of the tool and collaborate on creating backups (revisions).

3 A DESIGN OF A NEW TOOL

Project *Sandglass* (developed by the author within his bachelor's thesis) is a tool similar to *etckeeper*. It eliminates problems present in the solutions mentioned in Section 2 and introduces some new features. However, the workflow is different—the main concept is depicted in the Figure 1.

The tool is designed to be used both passively and actively. Forenamed solutions require system administrator's interaction. Nowadays, more regular and unskilled users are using GNU/Linux and modify system settings. The tool should give them a possibility to revert changes without creating a backup before. Such a functionality is achieved by periodic-time-based backups. To disable backups while applying some changes on the configuration files, *Sandglass* watches the modification progress and creates a backup after the modification is finished (i.e. after a certain delay from the last change). Another feature of the tool is that it is able to cooperate with a package manager like *yum* used in Fedora distribution.

On the other hand, system administrator can tell *Sandglass* that some changes are going to be applied. All of these changes can be annotated and also separated from regular backups. This will create a branch-structured history (similar to feature-branches known from version control systems [3]). Another benefit of the active approach is a cleaner final history.

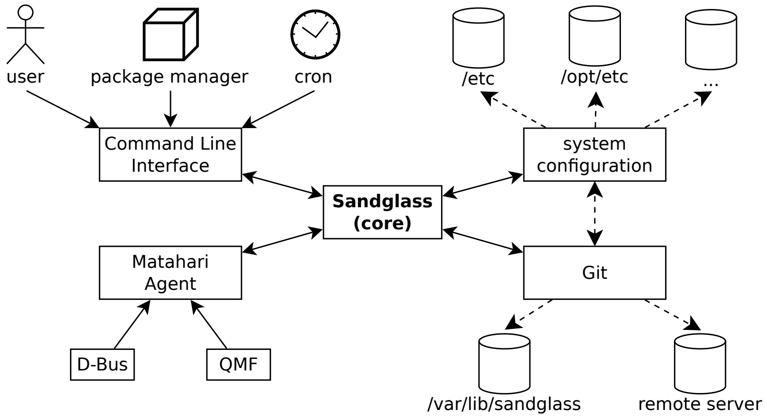

Figure 1: A concept of operations within the *Sandglass*.

Sandglass uses *Git* as the versioning backend. *Git* was chosen for its distributiveness, high performance, and security. Another advantage of *Git* is its separation into a bunch of small tools. This allows *Sandglass* to work with internal *Git* objects easily, flexibly, and very powerfully. System administrator does not have to be familiar with *Git*. *Sandglass* provides a few simple commands for versioning control of configuration files.

All necessary data are stored in compliance with *Fedora Packaging Guidelines* [4] (and so the *Filesystem Hierarchy Standard*) in `/var/lib/sandglass`, thus the problem of putting any metadata files into directory with configuration is removed.

Sandglass has an ability to submit the history to another server. The server can store history from multiple machines. This can be used for comparison of history and configuration between these machines. Configuration files can be also modified on the server and pushed back to the client.

Automatic operations are invoked via *cron* (time based scheduling) or by package manager events.

An administrator can run all operations using command line interface. Client-server communication is performed by executing remote commands via SSH connection. Alternatively, *Matahari Agent* [5] will be implemented, which will allow integration of *Sandglass* into other system configuration tools as a part of currently developed Fedora feature, *Matahari Based Fedora Management and Configuration Infrastructure* [6].

4 CONCLUSION

Sandglass should provide easy to use tools for versioning of configuration files. Although the tool is primarily designed for Fedora and Red Hat Enterprise Linux, the interface is designed to be easily extended to support other distributions and package managers. Additional information and development progress will be published on *Sandglass Project Website* [7].

ACKNOWLEDGEMENT

I would like to thank Aleš Smrčka, my bachelor's thesis supervisor, and Radek Vokál from Red Hat Czech for their valuable suggestions and remarks.

REFERENCES

[1] Filesystem Hierarchy Standard Group. *Filesystem Hierarchy Standard*. Freestandards.org, 2004-01-29. <http://www.pathname.com/fhs/pub/fhs-2.3.pdf>.

[2] J. Hess. *Homepage of etckeeper*. 2007 [quoted 2011-02-27]. <http://kitenet.net/~joey/code/etckeeper/>.

[3] V. Driessen. *A Successful Git Branching Model*. 2010-01-05 [quoted 2011-02-27]. <http://nvie.com/posts/a-successful-git-branching-model/>.

[4] T. Callaway. *Fedora Packaging Guidelines*. 2011-02-04 [quoted 2011-02-27]. <http://fedoraproject.org/wiki/PackagingGuidelines>.

[5] A. Beekhof. *Agent*. *Matahari Project Homepage*. 2011-02-08 [quoted 2011-02-27]. <https://github.com/matahari/matahari/wiki/Agent>.

[6] J. Řezník. *Matahari Based Fedora Management and Configuration Infrastructure*. 2010-03-22 [quoted 2011-02-27]. <http://fedoraproject.org/wiki/Features/FMCI>.

[7] J. Včelák. *Sandglass Project Website*. 2010-02-27 [quoted 2011-02-27]. <http://jvcelak.fedorapeople.org/sandglass/>.