The Backup/Restore Utility Design Document

CS-307 System Practicum

-:Team Members:-

Indresh kumar B15317

Avinav Sanyal B15211

Avnish kumar B15109

Pramod Jonwal B15121

Revision History:

Version	Date	Author(s)	Description
v1.0	14/05/18	Indresh,Avinav,Avnish,Pramo	Initial version

		d	

Table of Contents

1 Introduction

This utility will provide user with provision to backup data on local hard disk, external hard disk as well as on a disk on a network. This utility will also provide features such as automatic backup, record of backups and full restore or a file restore. This utility uses rsync, cron and ssh.
Rsync is a copying/backup tool that supports copying of file locally as well as remotely. It also supports incremental backup.
Cron is a daemon that allows us to automate program and scripts. It is launched during the boot up. The sequence of tasks to be executed are stored in a files known as crontabs.
SSH is used to transfer files to the remote server.
This document briefly explains the design of backup/restore utility and it's intended audience.

1.1 Design Overview

Backup/Restore system is based on rsync utility which is commonly found on UNIX like systems. It helps in efficiently transferring and synchronizing files across a network. Clients would be able to backup their files/folders by assigning low/high priority to them. A global node which handles the request from different clients for backup/restore would help in keeping higher number of copies for a file with higher priority across multiple servers available as backup disks. This would guarantee high availability and reliability to clients during restore phase.

1.2 Intended Audience

This SRS document is intended for anyone who want a Backup/Restore utility like linux Backup/Restore. Specifically for professionals like software managers and designers, network administrator and also for the general tech enthusiastic people.

1.3 References

1. https://en.wikipedia.org/wiki/Rsync

2.https://www.digitalocean.com/community/tutorials/how-to-use-rsync-to-sync-local-and-remote-directories-on-a-vps

3.https://unix.stackexchange.com/questions/211595/linux-dump-which-folders-files-are-excluded-from-first-backup

2 Detailed Design

2.1 Architecture

Components

Rsync is used as a tool to synchronize the files on local machine, external hard disk or remote host across the network.

Tar is used to compress the data which user like to backup.

SSH server is used to make the connection and to communicate with the remote host in order to take backup of the data or restoring the data from the remote server.

Fswatch is a file change monitor that receives notifications when the contents of the specified files or directories are modified and we will use Fswatch to detect if any changes are made to the specified file.

Interfaces

The interface between all pairs of components is a standard python interface made with tkinter.

Algorithms and Data Structures

Data structures

1. **Dictionaries**
   To store backup id along with date, type and importance of backup.

Algorithm

```
1. Backup( Blocation, Ltype, SLocation, Btype){

        Call space optimization
        If type is local
            Call rsync to create backup of Slocation at Blocation (full or incremental
            depends on Btype)
            Call tar to compress this temporary backup
            Delete the backup
        If type is remote
            Establish SSH connection to Blocation
            Call rsync to create backup of Slocation at Blocation (full or incremental
            depends on Btype)
            In remote terminal
                Call tar to compress this temporary backup
                Delete the backup
   }

2. FullRestore (Bid, Slocation){

        If Bid is an incremental backup
            Get details of all the backups till previous full backup
            Uncompress all the targeted backups
            Call rsync to restore the backup with id Bid
            Delete the files after restoring keeping only the compressed files for space
            optimization
        If Bid is a full backup
            Uncompress the backup with id Bid
            Call rsync to restore the backup
            Delete the uncompressed file
}

3. **Restore**(Filename, Bid, Slocation){

        If Bid is a full backup
            Using script search for the desired file and show options
            Restore the selected file and save it in Slocation

        If Bid is a incremental backup
            Using script search for the desired file and show options
            Restore the selected file and save it in Slocation
        Else
            Recursive call with previous backup
}

4. SpaceOptimization(){

        If free disk is below threshold
            Delete the oldest least important backup (full backup and its incremental
            backups)
            Recursive call
}
```

2.2 External Data

Databases

1. A table to store client info like client username and client id.
2. A table to store backup location.
3. A table to store diskID, space available,etc.

Files

1. Configuration files containing Client id, path of folder/file and a backup/restore flag.
2. A file which keeps all the commands of rsync, tar and ssh.

2.3 Performance

The utility will be tested on different PC's with different type of connection (Ethernet, wifi and local). This will help in obtaining a vivid performance analysis of the utility. On each

PC no. of backups will be created in order to check the space optimization performance of the utility. The utility will be reviewed by stakeholder to rate the UX.

2.4 Test Scripts

Scripts will be written to automatically update timestamp of some files during testing to check the efficiency of incremental backup.

;