

Ханенко Олександр Андрійович

студент

Національний технічний університет України «Київський політехнічний інститут»

Медведський Андрій Миколайович

студент

Національний технічний університет України «Київський політехнічний інститут»

Ханенко Александр Андреевич

студент

Национальный технический университет Украины «Киевский политехнический институт»

Медведский Андрей Николаевич

студент

Национальный технический университет Украины «Киевский политехнический институт»

Khanenko O.

student

National Technical University of Ukraine “Kyiv Polytechnic Institute”

Medvedskyi A.

student

National Technical University of Ukraine “Kyiv Polytechnic Institute”

СИНХРОНІЗАЦІЯ ФАЙЛІВ І РЕЗЕРВНЕ КОПІЮВАННЯ ЗА ДОПОМОГОЮ RSYNC СИНХРОНИЗАЦИЯ ФАЙЛОВ И РЕЗЕРВНОЕ КОПИРОВАНИЕ С ПОМОЩЬЮ RSYNC FILE SYNCHRONIZATION AND BACKUP WITH RSYNC

Анотація. Мета даної роботи – розглянути можливості утиліти Rsync як засобу для синхронізації файлів та створення резервних копій на віддалених машинах.

Ключові слова: синхронізація, Rsync, резервне копіювання, віддалений доступ, SSH, Linux, GUI.

Аннотация. Цель данной работы – рассмотреть возможности утилиты Rsync как средства для синхронизации файлов и резервного копирования на удаленных машинах.

Ключевые слова: синхронизация, Rsync, резервное копирование, удаленный доступ, SSH, Linux, GUI.

Summary. The aim of this work is to consider the possibilities Rsync utility as a tool for file synchronization and backup on remote machines.

Key words: synchronization, Rsync, backup, remote access, SSH, Linux, GUI.

Вступ

Одним з найбільш зручних інструментів, які встановлюються на Linux та інші *NIX системи є Rsync. Як випливає з назви, Rsync використовується для синхронізації файлів на локальних і віддалених машинах. Rsync використовується для копіювання файлів на віддалені системи і навпаки, або для резервного копіювання на локально встановлені жорсткі

диски. На перший погляд користувача, Rsync може здаватись важким та незрозумілим засобом, однак він є одним з дійсно необхідних інструментів.

Синхронізація файлів за допомогою Rsync

Загалом Rsync має одне застереження. Він є ефективним інструментом для переміщення даних з одного комп'ютера на інший, однак при неправильному

використанні, він є інструментом перезапису або видалення файлів. Відновлення файлів у Linux — дуже громісткий процес.

При запуску інструменту вперше, користувач має можливість скористатись опцією `-n` або, якщо перевага віддається довгому запису параметра, — `dry-run`. Виконання цієї команди призводить до пробного запуску без будь-яких реальних змін.

Для особистого використання, користувачі часто вибирають `Ifolder` або `Dropbox` для простих резервних копій. Ці інструменти відповідають вимогам клієнтів, серед яких:

- резервне копіювання;
- віддалений доступ;
- інші.

Однак ці сервіси відрізняються від `Rsync`. `Rsync` здебільшого використовується, щоб виконати резервне копіювання або складні операції синхронізації. Він містить великий стек функціоналу, який GUI інструменти не підтримують або підтримують, але роблять це погано. Прикладом такої функції є синхронізація файлу з домашньої директорії на локально примонтований USB накопичувач. Хоча й у GUI програм є також переваги: можливість загального доступу до файлів іншим користувачам чи групам, просте видалення файлів у папці (синхронізовано з віддаленою машиною).

Принциповим у `Rsync` є те, що файли копіюються не повністю, а лише на частину, яка відрізняється. Припустимо, у нас є два комп'ютери загального призначення `F` і `S`. Комп'ютер `F` має доступ до файлу `A` і `S` має доступ до файлів `B`, де `A` і `B` «схожі». Існує повільний канал зв'язку між `F` і `S`.

Алгоритм `Rsync` складається з наступних етапів:

1. `S` розбиває файл `B` в серію фіксованого розміру блоків, що не перекриваються розміром `L` байтів. Останній блок може бути коротше, ніж `L` байт.

2. Для кожного з цих блоків `L` обчислює дві контрольних суми: слабку 32-бітну контрольну суму і сильну 128-бітну за алгоритмом MD4.

3. `S` посилає ці контрольні суми до `F`.

4. `F` виконує пошук, щоб знайти всі блоки довжиною `L` байт (з будь-якого зсуву, а не тільки кратні `L`), які мають одну і ту ж слабку і сильну контрольні суми як одного з блоків `B`. Це може бути зроблено за один прохід дуже швидко, використовуючи нижче особливу властивість прокату контрольної суми.

5. `F` посилає `S` послідовність команд для побудови копії `A`. Кожна команда або являється посиланням на блок `B`, або буквенними даними. Буквальні дані передаються тільки для тих розділів, які не відповідають жодному з блоків `B`.

Кінцевим результатом є те, що `S` отримує копію, а тільки шматки, які не знайдені в `B` (плюс невелика

кількість даних для контрольних сум і блокових індексів), передаються по посиланню. Алгоритм зводить до мінімуму вплив затримки лінії зв'язку.

Найбільш важливі деталі алгоритму є прокат контрольної суми і пов'язані з ними кілька альтернативних механізмів пошуку, які дозволяють реалізувати дуже швидкий пошук контрольних сум.

Більшість дистрибутивів Linux має `Rsync`, встановлений за замовчуванням. Якщо немає, то варто використати менеджер пакунків дистрибутиву для установки `Rsync`. Наприклад, на OpenSUSE потрібно ввести:

```
sudo zypper in rsync
```

Наприклад, щоб скопіювати файли з домашньої директорії на диск USB:

```
rsync -avh --exclude="*.iso" /home/user/bin/ /media/diskid/user_backup/
```

`Rsync` дозволяє комбінувати опції після дефісу. Це один з пунктів, які роблять його зручним у використанні. В даному випадку після команди `rsync` чотири опції і один аргумент. Ключ (`-a`) говорить `Rsync` копіювати файли рекурсивно і зберегти групу і власності користувача, коли він копіює файли. Опція (`-v`) описує в терміналі дії `Rsync` більш детально. Це зручно використовувати для тестування команди, перш ніж додавати її в сценарій. При використанні цієї опції `Rsync` буде друкувати список змінених файлів, час відправки, обсяг даних і т.д. Однак кількість інформації буде обмеженою і вона буде неповною. І при потребі повністю вимкнути інформацію, варто передати ключ `-q`. Під час копіювання `Rsync` виводить інформацію, яка не є читальною для користувача. Для того, щоб зробити її доступною, варто передати також ключ `-h`.

Ще однією командою, вказаною в прикладі є `--exclude`. Вона зазначає, які файли повинні бути проігноровані, в даному випадку це файли з ізо розширенням.

Створення резервних копій

Як було описано вище, однією з ключових властивостей інструменту є можливість створення резервних копій. Зазвичай вони створюються на віддалених комп'ютерах, тому необхідно розглянути саме цей випадок.

`Rsync` може бути налаштований декількома різними способами для створення зовнішніх резервних копій, але варто розглянути найбільш практичний (також найпростіший і безпечний) метод тунельного `Rsync` через SSH. Більшість серверів і навіть багато клієнтів вже мають SSH, і він може бути використаний для резервного копіювання `Rsync`. Необхідно показати процес, щоб отримати одну машину Linux для резервного копіювання на іншу по локальній мережі.

Процес буде точно такий же, якщо робити це в Інтернеті. Варто лиш звернути увагу, що порт 22 (або будь-який інший порт, на який налаштований SSH), повинні бути спрямовані на будь-якому мережевому обладнанні на стороні сервера.

Для цього використовується та ж команда, що і для використання Rsync на локальному комп'ютері, але вона включатиме необхідні доповнення для тунелювання Rsync через SSH до сервера в своїй локальній мережі. Для користувача «user» підключення до «192.168.235.137» і, використовуючи такі перемикачі (-av -delete) ми виконаємо наступне:

```
$ rsync -av -delete -e ssh /Directory1/
user@192.168.235.137:/Directory2/
```

Якщо SSH прослуховування на якомусь іншому порту, а не 22, то необхідно буде вказати номер порту, наприклад:

```
$ rsync -av -delete -e 'ssh -p 12345' /
Directory1/ user@192.168.235.137:/Direc-
tory2/
```

Виконання резервного копіювання по мережі в значній мірі є таким же, як при створенні резервної копії на локальному рівні. Єдине, що змінюється, так це команда, яка використовується. Необхідно також зауважити, що присутній запит на введення пароля. Це для аутентифікації через SSH. Однак можна налаштувати ключі RSA, щоб пропустити цей процес і це в значній мірі спростить автоматизацію Rsync.

В Linux може бути використаний інструмент Cron, щоб автоматизувати виконання команд, таких як Rsync. Використовуючи його, є можливість запускати резервне копіювання щоночі, або так часто, як це потрібно. Для редагування файлу конфігурації Cron потрібно ввести:

```
crontab -e
```

Cron використовує такий синтаксис: хвилини, годину дня, день місяця, місяць року, день тижня, команду. Наступна команда буде запускати команду Rsync щоночі 22:00 вечора:

```
0 22 * * * rsync -av -delete /Directory1/
/Directory2/
```

Перший «0» вказує хвилини години, а «22» вказує на 10 годин вечора. Так як необхідно, щоб ця команда виконується щодня, потрібно залишити інші поля із зірочками, а потім вставити команду Rsync.

Є кілька різних способів, за допомогою яких можна шифрувати резервні копії Rsync. Найпростіший спосіб встановити шифрування на самому жорсткому диску (той, де будуть зберігатись копії). Інший спосіб для шифрування файлів перед їх відправкою на віддалений сервер. В обох випадках Rsync виявляється одним з найбільш ефективних і універсальних інструментів резервного копіювання на сьогоднішній день, і навіть простий Rsync сценарій може врятувати користувача від втрати даних.

Література

1. Get to Know rsync [Електронний ресурс] — Режим доступу до ресурсу: <https://www.linux.com/learn/get-know-rsync>. — Дата доступу: 05.06.2016.
2. How to Use rsync to Backup Your Data on Linux [Електронний ресурс] — Режим доступу до ресурсу: <http://www.howtogeek.com/135533/how-to-use-rsync-to-backup-your-data-on-linux/>. — Дата доступу: 28.05.2016.