rsync Backup

rsync Backup

rsync Backup

Evaluating rsync Backup for Servers

How Does rsync Work?

Computing Deltas with Check Sums

rsync is an efficient method for synchronizing a set of files over the network. rsync can compute the differences in two file trees efficiently using an ingenious specialized method of computing deltas using checksums called the rsync algorithm. Deltas are the parts of files that have changed from one version of a file set to another. rsync defines deltas as variable length blocks or chunks of data.

Using check sums to determine the differences between files is nothing new. What rsync added is the idea of a weak and strong checksum. For each file rsync breaks the file into small fixed size chunks of data called blocks. Each block of all files are read into memory where rsync computes the weak check sum (adler-32) of the block. The adler-32 check sum requires very little CPU time to compute. The weak check sum of a new block in a file is compared to the weak check sum of a block in the older version of a file. If the weak check sum is different the block has changed and is copied from the source to the destination.

By its design the adler-32 check sum is "weak" meaning there is a fair chance two blocks of data that are Not the same could actually compute to the same check sum. This would be bad as if this happened rsync would not detect that the block changed and the file would be corrupt. To overcome this matching weak check sums are verified with a strong check sum (MD4) which is almost impossible to generate a match on two different blocks. If the strong check sum is the same then the block is determined to not have changed. If it is different the block is updated in the destination file set and is determined to have changed.

What rsync Does Well

The whole idea of the rsync algorithm is to copy as little data across a slow network link as possible to update an older version of a set of files on another computer. rsync does a great job of this as it's guaranteed to only send deltas or different parts of files that have actually changed.

In doing this rsync aims to reduce CPU time needed to compute check sums. This is accomplished as rsync avoids computing the more expensive string check sum (MD4) as much as possible. And remember deltas are just what parts of the file changed. This is a great idea and works very well.

rsync Backup Resources

rsync backup recipes - http://www.mikerubel.org/computers/rsync_snapshots/

incremental backups with rdiff - http://www.gnu.org/savannah-checkouts/non-gnu/rdiff-backup/

rsync incremental backup script - http://www.rustyparts.com/ribs.php

official rsync examples - http://samba.anu.edu.au/rsync/examples.html

rsync backup documentation written by Kevin Korb - http://www.sanitarium.net/golug/rsync_backups.html

History of rsync

Did you know the author of rsync Andrew Tridgell did not intend rsync to be used as a backup application? Instead he developed rsync for general purpose file transfer and mirroring over slow WAN links. Tridgell explains in his PhD thesis paper that while the rsync algorithm would be highly useful for optimizing tape backup applications the rsync application itself is not intended to be used for backups.

rsync Challenges

Must Read All Data on Disk to Compute Deltas using Checksums

The real challenge as any server administrator knows is that in a modern server CPU time is usually plentiful while disk I/O is very precious. rsync must read all of the data on disk or in the file set to compute the checksums. This means the rsync operation takes longer and longer the more data there is no matter how little data has actually changed. This is why your server I/O wait time skyrockets along with waiting processes and load average when rsync runs on a large data set.

For more information see: Why Are Server Backups So Painful?

rsync is File Based and Not Block Based

Another challenge is that rsync works at the level of files and directories like any other program. This might seem confusing considering the previous paragraphs discussed how rsync divides files into blocks. What this means is that rsync examines each file and must create an index of

the path to every file and directory on the server. For a server with a fairly large number of files this can be an enormous task all on its own. It's also the reason people often have problems with rsync's consumption of large amounts of memory on larger servers. This is different from block based backup software that bypasses the file system to read low level disk or volume blocks.

rsync Keeps only One Copy of Data

On its own rsync keeps only one copy of the data. A basic feature of backup and archiving is the ability to keep multiple historical archive points as defined by some policy. There are some very clever scripts that combine rsync with Linux file system hard links to create an effective system for having multiple rsync archive points. A great resource for how to script incremental backups with rsync is http://www.mikerubel.org/computers/rsync_snapshots/

rsync Is Not Able to Get a Consistent Copy of Data

If you use rsync to copy or update a data set you know for sure is not being written to by any other users then your rsync copy is completely consistent. This is typically Not the case in practice. Typically when rsync is used for backup purposes it is run on a set of files that are being written to by other users and programs on the server. If a file is written to why rsync is reading it the file will likely be corrupt. If files are being added or removed from the server while rsync is running rsync will see some of the changes and miss others causing the file set to not be consistent.

The longer rsync needs to run for the more this is a problem. Consider how many changes are made to a server while rsync runs for 2 or 4 or 8 hours? What writes and changes made it into the backup? Which ones didn't?

How is Continuous Data Protection Different?

Continuous Data Protection as a technology is different in three primary ways.

CDP Uses a near-Continuous Method of Computing Deltas

As you read above a major drawback to rsync is the time and Disk I/O work needed to compute deltas using check sums. Continuous Data Protection solves this problem with a proprietary device driver that can track or see changes to the disk or volume at a very low level while the system is running.

Imagine if you will that the data on your disk is kind of like the earth's surface. Most of the earth's surface is not changing. At the same time forests are being cut down. Buildings and roads constructed. Houses being re-painted and changed. Every time rsync needs to make a map of your files or hard disk it must walk the earth to identify all the little changes. This takes forever of

course. Alternatively Continuous Data Protection works as if it was a giant database that got instantly updated every time a change is made. Ask it at any time what the map looks like and it has an instant up to date copy!

For more information see: Computing Deltas - near-Continuous (CDP)

CDP Is Block Based and Works at the Level of the Linux Kernel

Continuous Data Protection is block based bypassing the file system. It is not concerned with the number of files. It does not index file on the file system. It only reads deltas. It knows what the deltas are before the backup operation starts because of the near-Continuous method for computing deltas.

CDP Creates a Point-in-Time Snapshot of the Volume

In order to read data below the file system at the block level their must be a perfectly consistent point-in-time snapshot of the disk all while the system is running. R1Soft uses Windows Volume Shadow Copy to create snapshots on Windows servers and a proprietary block device snapshot driver in Linux. The Linux block device driver can be inserted into an existing block device and creates point-in-time copy-on-wright snapshots.

This means that the backup appears just as your system did at that point in time. For example your very First backup with R1Soft will take a long time as it must copy all of the data the first time of source During this time on a very large data set it could take hours or even days for users sending data over slow Wide Area networks. During all of that time the Linux and Windows CDP Agents are reading data from a consistent point-in-time snapshot that looks exactly like the disk did when the backup started.

CDP Uses Virtual Full Backups for Unlimited Recovery Point Archiving

CDP works by only doing a full backup once. Once and only once. From then an every backup operation is really a "synchronization" where deltas are copied. Through proprietary technology called the "Disk Safe" R1Soft is able to store deltas in a highly efficient manner. Only block level deltas are ever stored. Furthermore they can be optionally compressed. Each backup or synchronization appears as a virtually Full Backup even though it's made up of only deltas. For more details see: Backup Method - Virtual Full Backup

Technology Comparison

	rsync	R1Soft CDP
Backup Software Technology Category	Legacy	Continuous Data Protection

Backup Method Used	Incremental (only one copy is kept)	Virtual Full
Data Lost In Disaster	Days	Minutes
File Or Block Level	File Level	Block Level
Method for Computing Deltas	Check Sums	CDP
Backup Window Length	Hours or Days	Minutes when CDP is in Sync
Online Backup (Snapshots)	No	Yes Integral Part of the Product
Bare-Metal Restore	No	Yes
Backup Window Critical Path	= Time to Read All Data from Disks (needed to compute checksums)	= Only Time to Read Deltas
Date Invented	1999	2005
Core Technological Concept	Copy small set of files Use on slow Wide Area Network links Written by Andrew Tridgell Project for PhD Thesis Paper	Minimal Backup Window Little or No Load on Production Server Virtual Full Backups Continuous Data Protection Proprietary Device Drivers Web Interface Easy of Deployment Manageability Data Centers Multi-User

Linux Server Data Protection

	rsync	R1Soft CDP
Easy To Use Web Interface	No	Yes
Point-in-Time Snapshots	No	Yes (does NOT require LVM)
Bare-Metal Restore	No	Yes

Real-Time Backup (backup without interrupting your work)	No	Yes
Hourly or Minutely Backups	No	Yes
Backup Archiving	No	Yes
Virtual Full Backups	No	Yes
Significantly Reduce Disk I/O Load with Linux Kernel Level CDP	No	Yes

Windows Server Data Protection

	rsync	R1Soft CDP
Easy To Use Web Interface	No	Yes
Point-in-Time Snapshots	No	Yes
Bare-Metal Restore	No	Yes
Real-Time Backup (backup without interrupting your work)	No	Yes
Hourly or Minutely Backups	No	Yes
Backup Archiving	No	Yes
Virtual Full Backups	No	Yes
Significantly Reduce Disk I/O Load with R1Soft Windows CDP Volume Filter Driver	No	Yes
Windows Volume Shadow Copy Service Integration	No	Yes
Copy Windows Files from CIFS / Samba Network Share to Linux	Yes	No

Special Features for Hosting Service Providers

	rsync	R1Soft CDP
Easy To Use Web Interface	No	Yes
Multi-User System for Backup and Restore with Granular Permissions	No	Yes

Customer self-managed backup and restore	No	Yes
Quotas for Backup Storage	No	Yes
Hosting Control Panel Integration	No	Yes
Billing API for Service Providers	No	Yes