Computing Deltas - near-Continuous (CDP)

Computing Deltas - near-Continuous (CDP)

Methods for Computing Deltas in Backup Applications

While there are hundreds of different backup applications all of them use one of several known methods for computing Deltas.

Deltas are simply defined as the data that has changed since the last backup run. Defining it any further than that depends on how the backup application computes deltas. A delta could be a raw disk block, a variables length portion of a file or even a complete file depending on the method.

near-Continuous Delta Method (CDP)

The most efficient method for computing Deltas is the near-Continuous or CDP method. R1Soft happens to be the only example of a near-Continuous Deltas method for both Windows and Linux platforms. The near-Continuous method works by using a Disk Volume Device Driver. The device driver is situated between the file system (e.g. NTFS) and the Disk Volume (e.g. Logical Disk Volume 1).

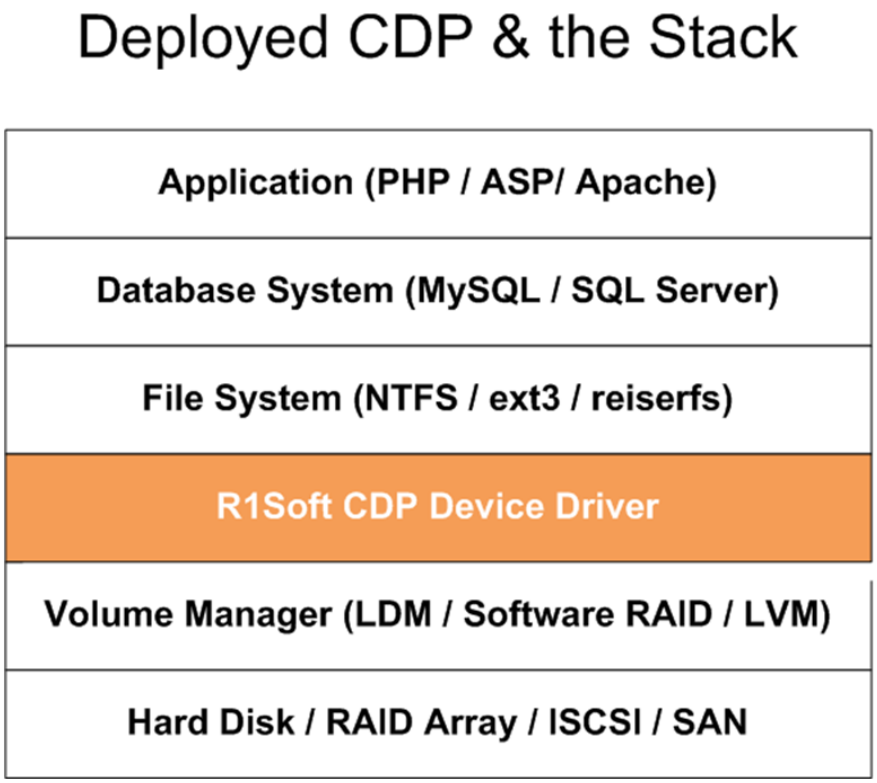
By locating a device driver between the file system and raw Disk Volume the application is able to identify changed Disk Blocks in real-time without any performance impact. Its really quite a simple concept. In Windows this kind of Device Driver is called an Upper Volume Filter Driver. R1Soft's Linux CDP implementation also uses a device driver. Linux does not have an official filter driver API though the form and function is very similar to the Windows CDP driver.

"Why spend hours reading data from the Disk just to compute Deltas when you can watch them happen for free?", says David Wartell, R1Soft Founder.

With the near-Continuous method of Delta computation a fixed length block size is used that in practice usually corresponds to the file system block size. Typically this fixed block size is 4 KB but can vary in different environments or implementations. As writes to the Disk are observed the block number that was changed is recorded in a specialized in-memory data structure.

R1Soft Linux Agents versions 1.0 employ a bitmap for this purpose where a region in memory uses 1 bit to describe the state of a disk block. Commonly bitmaps are used in image file formats. With a 4 KB block size there are 26,214,400 Disk Blocks per 100 GB of Disk Volume size. That corresponds to 26,214,400 bits or 3,276,800 bytes (3.125 MB) of memory to track all Deltas made to 100 GB of raw Disk capacity.

R1Soft 2.0 Windows Agents and later use a new proprietary data structure developed by R1Soft for tracking deltas. This new data structure is based on a Tree so that in the average case only 200 or 300 KB of memory is used to track all Deltas per 100 GB of raw Disk capacity. R1Soft is making this new more efficient data structure available to its Linux CDP technology with the release of Continuous Data Protection Server 3.0.



near-Continuous (CDP) Change Tracking Overhead

Changed Tracked in Memory

- R1Soft CDP 2.0 Change Log - 3 MB of memory used per 100 GB of raw Disk Volume capacity where CDP is enabled
- R1Soft CDP 3.0 - 200 - 300 KB on average per 100 GB of raw Disk Volume capacity

Disk I/O overhead Caused by CDP Change Tracking

- So small it's Not Measurable
- No Delay to I/O
- Windows and Linux kernel does very Little extra work (few dozen extra Assembly Language Operations)

Does near-Continuous (CDP) Deal with a Server Rebooting? (deltas are tracked in memory)

Yes a Reboot or Crash Automatically Triggers an Integrity Check to Re-Sync CDP

- Secure Random Key is Kept in Windows and Linux Kernel memory.
- Copy of this secure key is stored in the data repository with each synchronization.
- At the start of each synchronization operation the key in the data repository is compared to they key in kernel memory. If the keys differ then there was a reboot or crash of the server and an integrity check is required to re-sync CDP.
  Re-Syncing CDP
- Requires full block scan of Disk Volume and fails safe to checksum methods of computing Deltas since change tracking data structure is lost on reboot.
- Compares MD5 check sums of each used Disk Block to checksum in last completed recovery point synchronization.
- Only deltas are sent to network and stored.
- After Re-Sync (Integrity Check) CDP is back in Sync and near-Continuous Delta method is used.

Overview of near-Continuous (CDP) Delta Backup Method

Requires Time Consuming Walk of Entire File System Tree to Compute Deltas	No
Delta Granularity	Block
Accuracy in Identifying Changes	Perfect
Disk I/O Impact for Small Files	Only Deltas when CDP in sync
Disk I/O Impact for Large Files	Only Deltas when CDP in sync