

# rsbackup decay pruning

Richard Kettlewell

2015-09-12

## Abstract

Pruning policies are provided with a collection of backups with ages counted in days. The **decay** pruning policy is designed to keep all backups up to a (presumably small) limit and then thin out any older backups and keep older backups at a rate which decays over time. This article describes and justifies the implementation.

## Parameters

The policy has the following parameters:

Name	Value	Description
decay-start	$p \geq 0$	Age within which to keep all backups
decay-window	$w > 0$	Size of the decay window
decay-scale	$s > 1$	Scaling of the decay window.
decay-limit	$l > 0$	Limit beyond which to keep no backups.

## Description

### Preliminaries

An initial time window is preserved (i.e. not pruned) by subtracting  $p$  and discarding from further consideration any backup with an age  $a \leq 0$ .

Any backups older than  $l$  are pruned unless this would cause there to be no backups left whatsoever. The presence of a single backup, inside or outside, the initial time window is sufficient to cause ancient backups to be pruned.

### Decay Buckets

With these two issues dealt with, the remaining backups are organized into buckets  $B_n$  of size  $ws^n$  for integer  $n \geq 0$ . This is illustrated as follows:

$$a \in B_n \iff \begin{cases} 0 < a \leq w & (n = 0) \\ w < a \leq w + ws & (n = 1) \\ w + ws < a \leq w + ws + ws^2 & (n = 2) \\ \dots & \dots \end{cases}$$

In general,

$$a \in B_n \iff R_{n-1} < a \leq R_n$$

where:

$$R_n = w \sum_{i=0}^n s^i = \frac{w(s^{n+1} - 1)}{s - 1}$$

so:

$$\begin{aligned}
R_n < a \leq R_n &\iff s^n < \frac{(s-1)a}{w} + 1 \leq s^{n+1} \\
&\iff n < \log_s \left( \frac{(s-1)a}{w} + 1 \right) \leq n+1 \\
&\iff n = \left\lceil \log_s \left( \frac{(s-1)a}{w} + 1 \right) \right\rceil - 1
\end{aligned}$$

Having done this, for any bucket containing more than one backup, all backups except for the oldest are pruned. The reason that the oldest is kept rather than the newest is that otherwise a volume that was reliably backed up daily would never have any backups survive into  $B_2$  (or even  $B_1$  if  $w > 1$ ).