

Algorithm to calculate differences between two strings

To implement the delta snapshot technique, that is, after the first snapshot we need to only calculate the differences between the data (strings).

1. We have taken two strings OLD and NEW. OLD represents the first snapshot and NEW represents the actual data.
2. We divide the OLD string into blocks of a fixed size(say bsize= 4) and calculate checksum for every block using the given formula:
$$a(k, l) = \sum (X_i) \bmod M, \text{ where } i = l \text{ to } k$$
$$chk[k/bsize]=a(k,l)$$
$$len=\text{length of } chk$$
3. We have send the checksum for every block to the rch(NEW,chk,bsize,len).
4. In the rch() we parse the NEW string and calculate its checksum as follows:-

initially flag=0;

case 1: if(flag==0)

$$a(k, l) = \sum (X_i) \bmod M, \text{ where } i = l \text{ to } k$$
$$v=a(k,l)$$

case 2: if(flag==1)

Then we calculate the checksum using the previous checksum value(rolling checksum).

$$a(k+1, l+1) = (a(k, l), X_k + X_{l+1}) \bmod M$$
$$v=a(k+1,l+1)$$

5. Now, we will maintain a recovery string which will contain block references to the data common between OLD and NEW and the "modified data" for the unmatched parts of the two strings.
recovery string(str) = references(matched data) + data(unmatched data)
6. We will match v with the entries in chk :

case 1: if v == chk[i], then

$k=k+bsize$

$flag=0$

push i into a queue, where i denotes the reference to the block matched between OLD and NEW

$str = str+i$

case 2: if v does not match any of the enteries in chk , then

$flag=1$

$str = str+NEW[k]$

$k=k+1$

7. So, finally we get the recovery string(str) , which can be used to update the OLD string using only the differences.