Published: 10/21/66

Identification

Secondary Storage Residence Metering
T. H. Van Vleck

Purpose

Secondary storage usage will be metered in units of word-seconds, that is, words of secondary storage kept for one second. If all files were static, this quantity could be determined by a process which made a complete hierarchy scan, reading the length of each file and charging for that length multiplied by the time since the last scan. There will be a system process performing this action at relatively long intervals, in order to insure that even inactive accounts are charged completely at the end of every accounting period, and to minimize the effects of system crashes. Section BO.4.04 describes the hierarchy scan process.

File activity introduces some additional difficulties: a file may be created and then deleted before any scan discovers it; the length of a file may be changed between scans; and files will be deleted. In order to retain precision in accounting without forcing the scan period to be so short that the system does little else, we must introduce an additional mechanism, triggered by file activity, which will give us an up-to-date measure of each active account's secondary storage residence. Section BO.3.06 describes only this additional metering mechanism.

Method

Whenever an account's secondary-storage residence (in words) for any device changes, we wish to meter to the account a number of word-seconds

```
Old_length * time_at_length
```

for that device, where "time_at_length" is the time since the last similar computation, and "old_length" is the account's previous total residence on that device. Then we wish to compute a new figure for total residence on the device, for use the next time metering is done, and to note the time of the computation so that we can compute the next "time_last_metered."

Calls with the necessary information will come from the file system at a time when page faults are not allowed, so the metering must be done in the Active Meter Table, a wired-down data base described in B0.3.07.

If multilevel storage management moves a file from one device to another, we must perform the same computation for the old device and the new one, and then subtract the length of the file from the total for the old device and add it to the total residence for the new device. In this case, the file system will inform accounting of the move before it is made; an account will never be charged twice for the same file as a move file and a "real" copy.

Implementation

1. File activation

When a file is activated, we must insure that an AMT entry exists for the account number specified in the directory branch, so that we will have some place to put the metering figures. When the entry for the file is made in the Active Segment Table, the following call will be made to accounting:

```
AMTindx = start_sgt_meter(account_number);
```

"account_number" is the number contained in the directory branch for the file.

"Start_sgt_meter" will:

1) search for an AMT entry for this account. if none exists, one will be created.

2) increase the use count of the AMT entry by one.

3) return an index into the AMT which points to the entry. This index is stored in the AST for use by later calls.

2. Segment length change

If the length of a segment is increased or decreased, the following call will be made to accounting from the file system:

```
Call meter_length(AMTindx, device, length_change)
```

where

AMTindx is the index returned by "start_sgt_meter"

device is the device identifier code for the device on which the file residers, as in a file branch. See Section BG.7.

Length_change is the signed change in length. The current file system implementation will only report length changes in multiples of 1024 words.

This call

1) finds and interlocks the AMT entry. If the entry is interlocked, loop and wait.

2) computes

a) time_at_length = time_now - time_last_metered;

b) meter(device) = meter(device) + residence(device) * time_at_length;

c) time_last_metered = time_now;

d) residence(device) = residence(device) + length_change;

3) unlocks the AMT entry and returns.

4. Segment move

If multilevel storage management decides to move a file from one device to another, the following call will be made to accounting just before the move is initiated:

```
call meter_move(AMTindx, old_device, new_device, length);
```

where

AMTindx is the index into the AMT for the file

old_device and new_device are device identifiers for the devices which the file used to reside on and now resides on, respectively.

length is the length of the file in 1024-word blocks.

This call

1) finds and interlocks the AMT entry specified by "AMTindx".

2) computes as in steps a, b, and c for "meter_length" for both "old_device" and "new_device".

3) computes

a) residence(old_device) = residence(old_device) - length;

b) residence(new_device) = residence(new_device) + length;

4) unlocks the AMT entry and returns.

5. File deactivation

Whenever an AST entry is deleted, the following call comes from the file system at a time when page faults are possible:

```
call stop_sgt_meter(AMTindx);
```

this call

1) updates the usage meters in the AMT entry for all devices.

2) decreases the use count in the AMT entry by one.

3) if the use count becomes zero, calls "update_accounting" to update the information to the Account Data Segment, and then frees the AMT entry.