Published: 10/31/66

Identification

The Multilevel Storage System Move Control Module
Gerald F. Clancy

Purpose

The multilevel storage system move control module provides primitives for device management.

Introduction

The multilevel move control module is entrusted with the sole responsibility of distributing all segments known to the file system among the available secondary storage devices. Whenever a segment is about to be created move control is asked to specify an initial resident device. Whenever a segment is activated by the file system (section BG.3.00) move control examines the segment's user access history and determines if the segment should be moved to a different storage device whose speed characteristics are more commensurate with the current access history. Move control also maintains the device disposition table (DDT) which contains a count of space currently used, space currently available and critical occupancy thresholds for all devices.

The following primitives are provided by the move control module and are privileged to hard core procedures and to the multilevel storage monitor (section BH.1.02).

1. move_advice

This primitive is used by segment control (section BG.3.01) and by the multilevel storage monitor in order to ascertain the advisability of moving some existing segment to a new secondary storage device. The caller of move_advice extracts the items from the branch defining the candidate segment, sets a pointer to that items structure and makes the following call:

```
call move_advice (itemsptr, newdev, impval, offsw);
```

In this call, itemsptr is a pointer variable which points to the aforementioned structure of branch items (section BG.8.02 describes these items in full detail). newdev is the returned device identification of a new device if the segment is to be moved. If no move is required newdev will be returned as zero. impval is returned as the importance value computed for the segment by move control. The computation of a segment's importance value is described later under the get_impval primitive. offsw

is the move-off-to-detachable-storage-allowed switch signifying, if ON, that the caller is willing to move the segment to detachable storage if required. When the caller of move_advice is segment control offsw is always OFF since that module cannot move segments from the on-line storage system. When the caller is the multilevel storage monitor offsw may be ON since that process is capable of moving segments to detachable storage. Note that as described in section BH.0, the storage monitor does not physically move segments to detachable storage but merely truncates the on-line copy to zero length and leaves information in the branch indicating where on detachable storage the segment may be found if a future retrieval is required.

The itemsptr pointer avails to move_advice all the current items found in a branch. Those that move control uses to arrive at its decision are the following:

a) The two storage limit parameters which were assigned by the user and are used to control the device residence limits of the segment.

   hilimit is an indication of the desired upper device bound

   lolimit is an indication of the desired lower device bound

b) The current segment length--length

c) The current maximum segment length--maxlength

d) The current activity count and the time when the current activity count was last computed--respectively actcnt and cnttime (The activity_count primitive describes the significance of these parameters.)

e) The storage device identification where the segment currently resides--device

f) The date/time the segment was last used--dtu

When move control receives this call it first computes an importance value for the segment. This value is reached by considering the user set storage limit bounds (hilimit and lolimit), the current length (length), the maximum length (maxlength), the current activity count (the value of actcnt tempered by the current time less cnttime) and the date/time-last-used (dtu). All of the above parameters except hilimit and lolimit serve as an indication of the

relative desirability of the segment in relation to its residence on a particular storage device. The importance value computation is accomplished by passing all the above parameters as arguments to the get_impval primitive (described later in this section) which returns a resultant value.

There exists in the DDT for each device, current maximum and minimum importance values which determine what segments that device will accept. The allowable range prescribed for each device is set by the storage monitor as described in section BH.1.02. If the importance value of the segment in question is compatible with its current device then no movement is necessary and newdev is immediately returned as zero. Otherwise, the fastest device which accepts segments with an importance value of impval is found and its identification returned. If a move off to detachable storage is required but offsw if OFF, (signifying that the caller, probably segment control, will not accept moves to detachable storage) the identification of the slowest on-line device is returned.

Even if it is ascertained from the computation of the segment importance value that movement is required, any of the following conditions will keep the segment on its current device.

1. The maximum length of the segment is greater than that size deemed to be the largest allowed on that device. Within the DDT, the maximum segment size is specified for each device.

2. A move off to detachable storage is indicated, offsw is OFF and the segment already resides on the lowest on-line device.

3. The proposed target is currently unavailable to the reception of new segments because its space used exceeds the threshold value specified in the DDT.

Whenever a segment fault occurs, the segfault primitive of segment control is invoked to activate the missing segment. After creating the needed active segment table (AST) entry (section BG.2.00), move control is called at move_advice to determine if the segment should reside on a different on-line storage device. If a move is to take place segment control places additional information in the AST entry to define the move file on the new device. Movement then takes place as a result of normal paging in and out of core (see section BG.4.00 for a more detailed description). move_advice is also called directly by the multilevel storage monitor to determine the advisability of moving a particular segment. That process can then

decide to initiate a segment move by calling segment control at moveseg specifying the device identification returned by its own call to move advice as the target device. moveseg then initiates a second call to move advice to verify the device request legitimacy.

2. assign device

Whenever a new segment is to be created an initial device must be specified to receive pages. Segment control makes the following call for this purpose.

```
call assign_device (itemsptr,device);
```

In this call itemsptr is a pointer variable which points to the structure of branch items. maxlength, the maximum segment length and hilimit, the desired upper limit of residence, are extracted from the branch items and are used in assigning a device identification which is returned as the value of device.

Normally, new segments are created on the fastest on-line device. If any of the following conditions exist, however, a lower device in the hierarchy must be specified.

a) The maximum segment length (maxlength) is greater than the size which the fastest device is willing to accept. The returned value of device will be the identification of the fastest device whose size is compatible with maxlength.

b) The fastest device is currently unavailable for the reception of new segments because of a surpassed capacity threshold. device is then returned as the identification of the fastest available device.

c) The user prescribed value of hilimit is such that the segment should not reside on the fastest device. The returned value of device will be the fastest device compatible with hilimit.

3. set criterion

This primitive is used by the multilevel storage monitor to change the range of acceptible importance values for a particular device.

```
call set_criterion (did, max, min);
```

In this call did is the identification of the device to be affected. The value of max replaces the current maximum importance value and min the minimum value in the DDT for the device specified by did.

4. get impval

In order to compute a segment's importance value, the following primitive is provided for use by the multilevel storage monitor and the move advice primitive described earlier.

```
value = get_impval (hilimit, lolimit, length, maxlength,
        dtu, actcnt, cnttime);
```

In this call hilimit, lolimit, length, maxlength, dtu, actcnt and cnttime are as described under the move advice primitive. value is the returned importance value.

The algorithm used to compute value is presented here as an initial implementation only. It will be subject to future modification and is not based on a great deal of Multics operating experience.

When seeking the proper device residence for a segment four major usage characteristics serve as an indication of its importance and desirability.

a) The current activity count (actcnt). actcnt is a measure of the segment's rate of page I/O activity. Each time an I/O request is issued for any page of the segment, actcnt is incremented by some constant and with the passage of time actcnt is decremented. (This activity algorithm is described later in this section as the activity count primitive.) When an importance value is to be computed, activity count is called with actcnt and cnttime as arguments. A new actcnt is returned which is the proper count adjusted to the current time. In the computation, value should vary proportionally to actcnt since greater activity indicates greater importance.

   The component of the resultant value attributed to activity count is limited to some maximum (Mac) in order to control the final value and so that actcnt exerts only a limited effect on the result. The activity factor is computed as follows:

$$fa(actcnt) = \begin{cases} N * actcnt, & actcnt \leq Mac \\ Mac, & actcnt > Mac \end{cases}$$

   In this function the constant N proportions the result from the range of actcnt to the desired limits of fa.

b) The span of time since the segment was last used (current time-dtu). Current time less dtu is a measure

of a segment's long range usage by system users. If (current time-dtu) is very large then the segment is not in great demand and this fact should exert some influence in the computation of the importance value.

In specifying the algorithm used to compute the influence of dtu on value it is desirable to 1) limit the component to some maximum (Mu) and 2) set a limit of (current time-dtu) beyond which the function results in a minimum influence.

If that cut-off is Cu then the following function displays the desired characteristics

$$fu(now\text{-}dtu) = \begin{cases} Mu - \frac{Mu}{Cu}(now\text{-}dtu), & (now\text{-}dtu) \leq Cu \\ 0, & (now\text{-}dtu) > Cu \end{cases}$$

c) The current segment length (length). The shorter the length of a segment the more desirable it is in terms of residence on a faster device and hence, importance value should increase for shorter segments. The component of value supplied by length must be limited to a maximum (Ml) and a cut-off length (Cl) set beyond which length minimizes its component of value. The following function is the length component.

$$fe(length) = \begin{cases} Ml - \frac{Ml}{Cl}(length), & length \leq Cl \\ 0, & length > Cl \end{cases}$$

d) The segment growth potential. (maxlength-length). Segments having small growth potential are desirable for residence on faster devices and hence the growth potential component of value should reflect this. By including growth potential in the importance value, segments with small length and large maxlength are discouraged from residing on the faster devices and hence cannot greatly expand and hasten device saturation. The influence of growth potential is limited to a maximum (Mp) and there is a cut-off value (Cp) beyond which influence on value will be minimized. The function is the following:

$$fp\ (maxlength - length) = \begin{cases} Mp - \frac{Mp}{Cp} * (maxlength - length) \\ \quad , (maxlength - length) \leq Cp \\ 0, (maxlength - length) > Cp \end{cases}$$

The final computation of value is the sum of the four components previously defined.

value = fa (actcnt) + fu (now-dtu) + fe (length) + fp (maxlength-length)

value is assured to always have a value between zero and (Mac + Mu + Ml + Mp). If this sum is constrained to the number of random 1024 word pages per minute of I/O service that the user is likely to receive from the fastest on-line secondary storage device (about 6000 for the drum) then a relation is established between importance values and the various physical devices available. Thus a range of I/O service in 1024 word pages per minute for each device dictates an honest importance value range used to assign devices (see above, move_advice primitive). Then, if the user-declared storage limit parameters, hilimit and lolimit, are likewise expressed in these same units, then hilimit and lolimit can be used to temper the final importance value to the user's specifications.

Therefore the returned value will be the result of the above computation unless value is greater than hilimit or less than lolimit. Otherwise value is returned as the value of the storage limit excluded in the computation.

5. device_distress

Whenever the total percentage of space used on any device (measured relative to the total space available) exceeds the designated threshold, the responsible device interface module (DIM) is obligated to announce the state by the following call:

```
call device_distress (did);
```

When this call is received, the appropriate DIM interrupt switch, specified by the device identification did, is set in the DDT to signify that the threshold has been reached. The multilevel storage monitor is then awakened if it is not already operating.

6. set_threshold

The following primitive is used by the storage monitor to alter the capacity overflow threshold for a device:

```
call set_threshold (did, new, old);
```

In this call did is the device identification of the device to be affected. new is a number expressed as a percentage whose value becomes the new capacity overflow threshold and old is the returned value of the previous setting.

7. activity_count

Located within each branch is an activity count parameter (actcnt) and an activity count date/time (cnttime). actcnt is a measure of the associated segment's rate of I/O access. Each time an I/O request is issued for a page of a segment activity_count is called to compute a new actcnt. This primitive also allows the get_impval primitive to compute the current activity for a segment given the present value, actnt and the time when the value was last changed, cnttime.

```
call activity_count (actcnt, cnttime, incsw);
```

In this call actcnt is the current activity count and will also be returned as the new value. cnttime is the date/time the incoming actcnt was last changed and will be returned as the current time. Incsw is the increment switch specifying whether (ON) or not (OFF) actcnt should be incremented after computing the current count. incsw is ON only if the caller is queue control in which case the segment in question has just been given I/O service.

When the call is received, actcnt is incremented only if incsw is ON after being decayed by an established decay rate (found in the DDT) times the span of time since the last computation of actcnt.

```
actcnt = actcnt - rate*(now - cnttime);

cnttime = now;

if incsw = "1"b then actcnt = actcnt + n;
```

In this algorithm rate is the predefined decay rate, now is the current time and n is the incrementing constant.

Figure 1 shows the block relations between the move module and the rest of the file system.

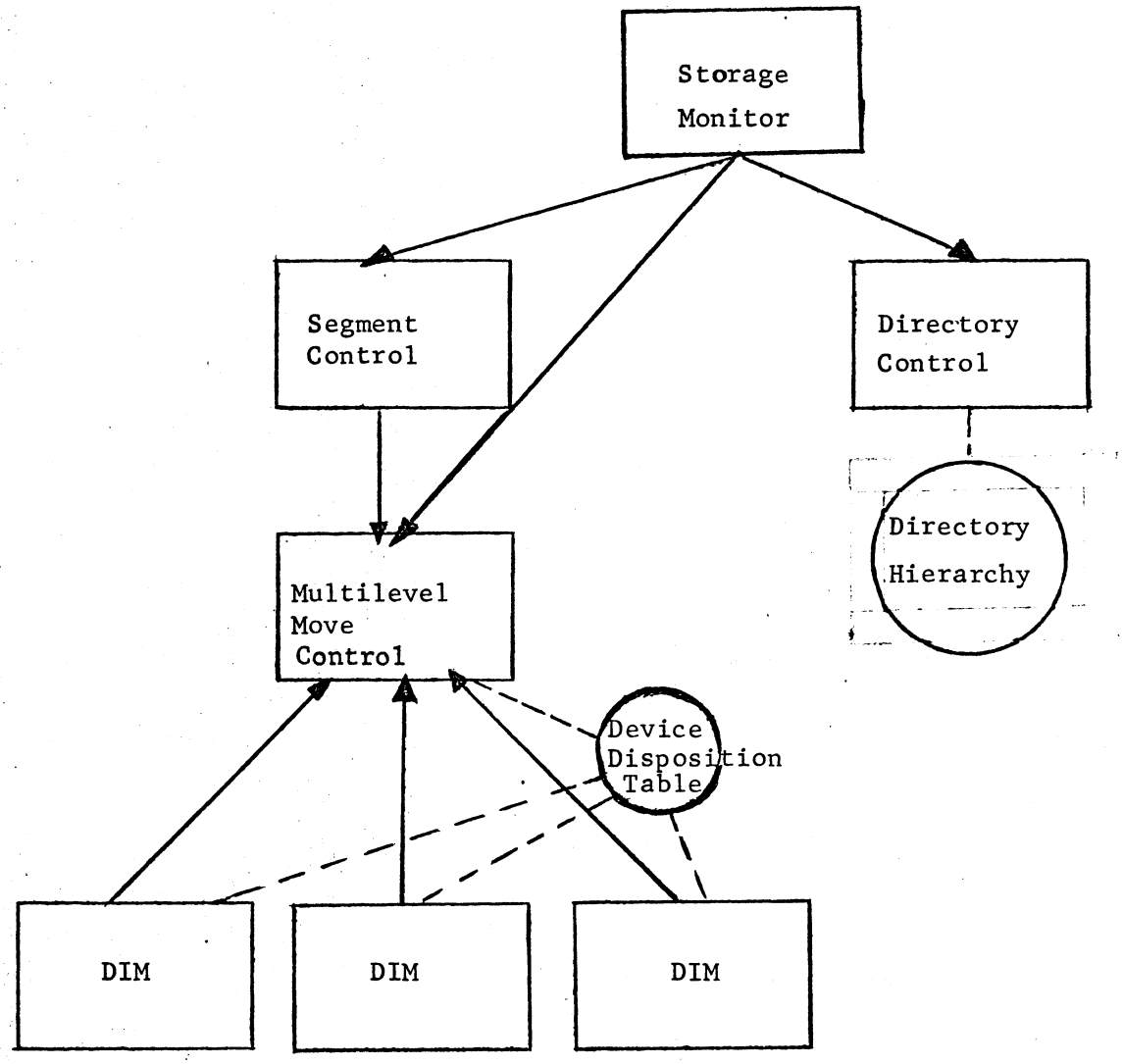

Figure 1

Block Diagram Showing Control Paths
Between the Multilevel System and the file system

Device Disposition Table

The device disposition table (DDT) contains information affecting the storage dynamics previously described and data relevant to each on-line storage device known to the file system. Its contents are the following:

1. Activity decay rate (rate)
2. Activity incrementing constant (n)
3. Maximum influence of activity count on the importance value computation (Mac)
4. Transformation factor of activity count into activity importance value component (N)
5. Maximum influence of date/time-last-used on importance value (Mu)
6. Cut-off of now-dtu beyond which date/time-last-used does not influence importance value computation (Cu)
7. Maximum influence of segment length on importance value (Ml)
8. Cut-off value of segment length beyond which length does not influence the importance value (Cl)
9. Maximum influence of growth potential on importance value (Mp)
10. Cut-off value of growth potential beyond which (maxlength - length) does not influence the importance value (Cp)
11. Number of secondary storage devices
12. Device table (one for each device)
    a) Device speed
    b) Availability switch
    c) DIM interrupt switch
    d) Space available on the device
    e) Space currently occupied
    f) Maximum resident segment length
    g) Importance value criterion
        1) Maximum setting
        2) Minimum setting
    h) Capacity overflow threshold
    i) Lock

The following is a brief explanation of each of the items present in the DDT.

1. Activity decay rate--this number determines how fast an activity count decays.

2. Activity increment factor--each time an I/O request is handled by queue control the segment activity count is incremented by this amount.

11. Number of devices--the number of distinct types of on-line devices known to the basic file system and the number of separate tables which follow.

11.a. Device identification--unique identification by which the device is known and talked about in the file system. Device hierarchy priority is implicit in the assignment of the identification (the lower the number the higher the priority).

11.b. Availability switch--this switch is set OFF by the multilevel storage monitor when the device is no longer able to receive new segments.

11.c. DIM interrupt switch--this parameter is set by the device distress primitive immediately before the multilevel relief process is called. It serves as an indication to that process of which device is becoming saturated and that the distress is being serviced.

11.d. Available space--this parameter is the apparent capacity of the device in units of 64 words (This may be less than the real capacity).

11.e. Space currently used--the amount of device storage now occupied in units of 64 words.

11.f. Maximum resident segment length--no segment whose maximum length is greater than this value is permitted residence on this device.

11.g.1. Maximun importance value criterion--no segment whose importance value is greater than this number may be moved to this device.

11.g.2. Minimum importance value criterion--no segment whose importance value is less than this number may be moved to this device.

11.h. Capacity overflow threshold--the capacity overflow threshold is the warning indicator to the DIM. When space currently used excedes this number the storage monitor must be awakened to move files from the device. The threshold is expressed as a percentage of the apparent available space.

PL/I Parameter Declaration

The following is the PL/I declaration of the previously defined parameters to the move control module.

```
dcl     actcnt      fixed bin (35),
        cnttime     bit(52),
        device      bit(4),
        did         bit(4),
        dtu         bit(52),
        hilimit     fixed bin (17),
        impval      fixed bin (17),
        incsw       bit(1),
        itemsptr    ptr,
        length      bit(12),
        lolimit     fixed bin (17),
        max         fixed bin (17),
        maxlength   bit (8),
        min         fixed bin (17),
        dev         fixed bin (17),
        newdev      bit(4),
        old         fixed bin (17);
```

PL/I Declaration for the Device Disposition Table (DDT)

The DDT and allocation area for the device tables are declared as follows.

```
dcl 1 ddt ctl (ddtptr),
    2 rate fixed,        /* activity count decoy rate*/
    2 n    fixed,        /* activity count increment
                            constant*/
    2 mac  fixed,        /* maximum activity component*/
    2 bn   fixed,        /* activity transformation factor*/
    2 mu   fixed,        /* maximum usage component*/
    2 cu   fixed,        /* date-last-used cutoff*/
    2 ml   fixed,        /* maximum length component*/
    2 cl   fixed,        /* length cutoff*/
    2 mp   fixed,        /* maximum growth potential
                            component*/
    2 cp   fixed,        /* growth potential cutoff*/
    2 nodevices fixed,   /* number of device
    2 tableptr  ptr,     /* pointer to devices tables*/
    2 tablearea area (A); /* device table allocation area*/
```

The pointer tableptr points to an array of device tables declared as follows.

```
dcl 1 devtables (ddtptr→ddt. nodevices) ctl(p),
    2 speed fixed, /* device speed*/
    2 availsw bit (1), /* availability switch*/
    2 intsw bit(1), /* interrupt switch*/
    2 spaceavail fixed, /* space available*/
    2 spaceused fixed, /* space used*/
    2 maxlength fixed, /* maximum resident segment length*/
    2 maximpval fixed, /* upper bound on importance value*/
    2 minimpval fixed, /* lower bound on importance value*/
    2 threshold fixed, /* overflow threshold*/
    2 lock bit (1); /* table lock*/
```