Published: 12/16/66

Identification

Incremental Dump Decision Module
S. H. Webber, G. F. Clancy

Purpose

The incremental dumping process fulfills three fundamental objectives of the overall backup scheme (see section BH.2.00). First, it provides for the dumping of the primary copies of all file system segments onto detachable storage; second, it insures that two more secondary copies of all segments are dumped sometime after the primary copies have been made; and, third, carries out the user directed dumping of specified hierarchy subtrees in a mutually consistent state. This section details the criterion applied to hierarchy data and the subsequent actions taken by the incremental dump decision module in order to fulfill the above goals. When read in context with section BH.2.00 this section yields a complete description of the incremental dumping process.

Introduction

The incremental dumping process operates continuously within the Multics system. Periodically it initiates a scan of the entire directory hierarchy tree and evaluates certain parameters within each directory entry in order to decide what data need be dumped. The dump module is notified whenever an entry or a primary copy of a segment must be dumped, whenever a secondary backup segment copy is to be produced, and whenever a user defined sub-tree is to be dumped in a consistent state.

The Incremental Dump Decision Module

The incremental dump decision module examines each directory entry read from the hierarchy by the scan module and decides what actions, if any, need be taken to fulfill its backup objectives.

Whenever an entry is passed, the following items are extracted from the entry structure and are used in applying the decision module's incremental dump criterion.

1. The date/time-entry-last-modified (dtem). This parameter exists for every directory entry (link, common access control list, branch) and is set to the date/time any of the data within the entry itself last changed.

2. The date/time-last-modified (dtm). This parameter exists within each branch and is set to the date/time when any data inferior to the entry last changed.

3. The date/time-last-dumped (dtd). This parameter exists within each directory entry and is the date/time when the entry itself was last dumped by the incremental dumper. (When changed by the dumping process it is set to the date/time the entry was read from the hierarchy by the scan module).

4. The branch type (dirsw). This switch is ON if the branch defines an inferior directory and is OFF otherwise.

5. The consistent dump switch (consw). This switch is ON in a directory branch if a user has specified that the sub-tree inferior to the entry is to be dumped in a mutually consistent state.

The incremental dump decision module is dynamically aware of all modifications taking place within the file system hierarchy tree. If a user alters a segment at some depth, that modification is reflected in the branch pointing to that segment by an updating of the date/time-last-modified in the branch data when a modified page is written onto secondary storage. In addition the date/time-last-modified is updated in all directory branches superior to the modified segment. A simple comparison of date/time-last-modified and the date/time-last-dumped produces an immediate picture of the backup status at each node of the hierarchy tree and also indicates whether a particular subtree need be scanned.

The incremental dump process copies onto detachable storage all newly created or modified non-directory segments and individual directory entries. However, in order to avoid excessive dumping of short-lived data, a latency period is set (M hours) which is the span of time an entry or non-directory segment must remain in a modified state in order to be dumped. M is measured relative to the date/time-last-dumped so that continuously modified data will eventually be dumped.

A modified entry is dumped if:

dtem > dtd and current time > dtd+M.

A modified non-directory segment is dumped if:

dirsw = "0"b and dtm > dtd and current time > dtd+M.

The consistent dumping of a hierarchy sub-tree is requested by consw being set equal to 1 in the directory branch defining the sub-tree. This condition is sent to the dump module which insures that consistent dumping has taken place and so records the fact (section BH.2.00).

The incremental dumping process dumps secondary backup copies of all segments whose primary backup copies have already been produced (see above) and which have been unmodified for some period of time. This time span (N days) is set to be long enough to insure that a secondary segment copy will not be dumped on the same physical unit of detachable storage as was the primary copy. A secondary copy of a non-directory segment is dumped if

1. dtd $\geq$ dtm,
2. the secondary copy has not yet been dumped, and
3. dtm < current time - N days.

The incremental dump decision module is capable of directing the subsequent path of the process through the hierarchy tree. Normally the next entry fetched by the scan module is the next lateral entry in the current directory. However, if the current entry is a directory branch defining an inferior subtree, then that subtree will be searched next by the scan module and will be directed to do so by the decision module if any of the following conditions exist.

1. The dtm>dtd in the current entry. This condition means that inferior undumped modifications exist and must be sought out.

2. The process is operating in secondary dump mode which means the secondary backup segment copies will be dumped if required. If secondary copies are to be dumped by the process, then all non-directory branches must be examined and, hence, a full tree search must be executed.

The following is a summary of the criterion applied by the incremental dump decision module.

1. Dump copies of all non-directory segments, links, branches and common access control lists which have been modified since they were last dumped and which have not been dumped in the last M hours.

2. Insure the recording of dumping in a mutually consistent state of all data within a given hierarchy sub-tree. Only that data will be dumped which is selected by normal application of other criterion.

3. If the dumper is operating in secondary dump mode and if the current entry is a non-directory branch and if the associated segment has been dumped since it was last modified and if it has been left unmodified for at least N days and if a secondary backup copy has not yet been produced then the dump module is instructed to effect the secondary dumping of the segment.

4. If the current entry is a directory branch and either operation is in secondary mode or the inferior directory has been modified since its entry was last dumped, then the scan module is instructed to scan that inferior directory next. Otherwise a normal return is taken and the inferior sub-tree will be skipped by the process.

The decision module uses one or more of five switches to communicate its decisions to the dump module. A full set of these switches is permanently attached to the entry copy in the position segment.

1. Entry switch (ESW). If the decision module turns this ON the dump module will dump a copy of the entry itself.

2. Segment switch (SDW). If this switch is turned ON the dump module will dump a copy of the non-directory segment defined by the associated entry.

3. Update switch (USW). The update switch is always set ON by the incremental dump decision module. If the update switch is ON for a given entry when encountered by the dump module then that entry's date/time-last-dumped is set to the date/time the current entry version was read from the hierarchy. In addition, if the update switch is ON for a non-directory branch when its associated segment is dumped, the position on detachable storage where the copy is dumped is written into the retrieval argument list in the branch. These two records of dumping are maintained only by the incremental dumping process.

4. Consistent dump switch (CSW). This switch is turned ON by the decision module for a directory branch entry if a user has specified that the contents of the hierarchy sub-tree defined by the entry are to be dumped in a mutually consistent state. The execution and recording of the consistent dump are described fully in section BH.2.00

5. Secondary dump switch (SSW). This switch is set ON for a non-directory branch whenever the two secondary copies of the associated segment are to be produced.

The incremental dump decision module is called by the following:

```
     call incremental_decision (dsw, psp, deeper);
```

<u>dsw</u> is the call-dump-module switch. This is set ON by the decision module only if at least one of the switches has been set. It is a signal, if ON, that some dumping operation is required for the entry and thus forces a call to the dump module from the scan module.

<u>psp</u> is a pointer to the base of the position segment which contains the items structure of the entry to be considered by the call to the decision module.

<u>deeper</u> this is the return to be made when the decision module realizes that the current entry is a directory branch and that the directory should be scanned. If a normal return is made, scanning continues at the current level. If the current entry is not a directory branch, but a return to <u>deeper</u> is made, no adverse action is taken.

The PL/I declaration of the arguments in this call is:

```
          dcl dsw bit (1),
              psp ptr,
              deeper label;
```