The Path to Improved IMAGE Detail Dataset Performance

Patrick Mullen
World-Wide Technical Support

Adager Corporation
Sun Valley, Idaho 83353-3000 U.S.A.
Tel. (208) 726-9100 *Fax* (208) 726-8191
patrick@adager.com *www.adager.com*

Input/Output (I/O)

Many articles have been written and many diagnostic tools have been developed to help identify sources of MPE/iX system performance degradation. Performance measurements are generally categorized by either CPU or I/O. In this article I address I/O performance problems that have been attributed to IMAGE detail datasets.

HowMessy and Adager

Even if no official performance analysis has been run on your system, the following techniques—which employ Robelle Consulting's *HowMessy* and Adager Corporation's *Adager*—can help identify and correct performance sinks. *HowMessy* is included in Robelle tapes and, thanks to Robelle's permission, it is also part of every Adager tape (*HowMessy.Library.Rego*).

In this article, I will help you interpret the report generated by *HowMessy* and employ the proper *Adager* functions to improve the performance of chained reads on detail datasets.

The two Adager functions that address detail performance problems are *DetPack* (or its equivalent, *Repack Dataset*) and *Change Primary.*

Follow the path

In order to use a *HowMessy* report to diagnose performance bottlenecks on detail datasets, you much first identify sets which have large numbers of entries.

Of course, size is relative from database to database. However, if in general there are relatively few entries in the detail, then that dataset should not be causing performance problems.

Having identified a detail dataset candidate, determine how many paths it has. (We focus on this aspect because this article deals exclusively with performance issues related to Chained Reads on detail datasets).

The *HowMessy* column "Search Field" contains the names of the paths in the detail. You can eliminate from your performance bottleneck consideration any paths possessing "Average Chain" lengths less than 2. These are paths that probably have one-to-one relationships with their associated masters. Such relationships have no inherent I/O bottlenecks reading records in a chained fashion. In such relationships, each time the detail is accessed via that search field only one detail entry will be retrieved. Therefore, you cannot gain any better chained-read performance from DetPacking these paths. In fact, it is possible to actually degrade performance on other paths by DetPacking a path with a one-to-one relationship.

The primary path

Any detail dataset with paths must have one path designated as the primary path. In the *HowMessy* report, the primary path is flagged with a "!" (*Adager* and *DBSCHEMA* use the same flag.) Assigning the primary path to a path with a one-to-one relationship with its master can mislead you when doing detail dataset performance analysis. For instance, even though *Adager* allows you to *DetPack* on any path, it's default is the primary path so that you can specify it by simply entering <return>.

DBLOAD, as well as some other database utilities, are only able to repack on the primary path. Many users may unwittingly choose this default and actually degrade performance in the process, because that path has a one-to-one relationship.

DetPack the path

Although *Adager* allows *DetPacking* along any path in the dataset, you may find it beneficial to change the primary path to the path that you know should be DetPacked regularly. Changing the primary path will also be helpful in the event that someone unfamiliar with the database needs to perform database maintenance. By properly designating the primary path, you take the guesswork out of determining which path should be DetPacked for future database administrators.

Use the Adager command *Change Primary* to change the primary path. This command will not affect the behavior of any of your applications. The primary path designation was implemented by the authors of IMAGE for the purpose of loading data into detail datasets during DBUNLOAD/DBLOAD, before Adager existed. (Fred White lobbied that it be called the default

path.) Adager merely exploits this IMAGE primary path designation as a means of offering the user a default in the *Chained* option of its *DetPack* command.

Chained DetPacking optimizes retrieval of detail records with identical search field values. The goal is to place all entries with the same search field values physically adjacent on a disk. This results in less I/O for retrieval on the path. In general, if users experience performance problems reading records it can usually be attributable to problems with the detail dataset. Likewise, if users experience performance problems adding records, it usually means there are some problems with a master dataset or perhaps a detail with a poorly designed sorted path. Apart from your users' input regarding the performance of IMAGE transactions, *HowMessy* is an excellent tool to determine which IMAGE performance issues exist.

Which path?

If you do not know which path should be designated as the primary path and you've already eliminated from consideration paths with one-to- one relationships, you should next eliminate any paths that have excessively large "Average Chain" lengths. Although these may cause performance bottlenecks in some cases, it is unlikely that users do chained reads on these search fields on a regular basis. These paths are often either batch-oriented in nature or represent a repeating value used to satisfy a search field that exists but is not used. In such instances, zeroes are often entered to satisfy the search field value.

You can suspect this if the *HowMessy* column values for "Maximum Chain" length and "Average Chain" length are large relative to the size of the dataset (*HowMessy* stops counting at 99,999). The "Maximum Chain" and "Average Chain" columns are tempered by the column "Std Dev" (standard deviation). In brief, the smaller the standard deviation the closer the "Average Chain" length is to the norm. You can confirm the usage of repeating, unused search field values with *QUERY*. The existence of these paths should be questioned and they should be eliminated for performance's sake if possible.

When to DetPack a path

Once you have identified which datasets and paths would benefit from DetPacking and have their primary paths so assigned, you can determine if the path is in need of DetPacking. The *HowMessy* report can tell you the number of I/Os it takes to retrieve the average chain length in the dataset's current state as well as how many it would take in its DetPacked state.

Since *HowMessy* was developed for the HP 3000 Classic system, in order to properly interpret its output on the HP 3000 RISC machines we must get one more piece of vital information not provided in the HOWMESSY report—the MPE/iX record length. To get the record length do a LISTF,2 of the database name and set number. The MPE/iX record size in words is the IMAGE block length, expanded to a multiple of 128 words.

Plug this IMAGE block length value—along with the column values for Average blocks, expected blocks and blocking factor obtained from *HowMessy*—into the following equations to determine the current I/O and the best obtainable I/O for the chains in the path:

Current I/O = (AverageBlocks*BlockLength/2048)/BlockFactor
Best I/O = (ExpectedBlocks*block length/2048)/BlockFactor

Current I/O shows how many disk reads it takes to retrieve the average chain length on this path in the dataset's current, unpacked state.

Best I/O shows how many disk reads it would take to retrieve the average chain length after the dataset has been DetPacked. Both measures presume none of the chain is in memory. Any value less than 1 must be interpreted as 1; the remainder of the retrieval will be members of the adjacent blocks up to a total of 2048 words. If the difference between these two values is significant, doing an Adager *DetPack* with the *Chained* option will improve the performance of chained reads on this path.

Packed enough?

HowMessy's "Elongation" column gives you the ratio of average blocks over expected blocks. In effect, this is the number of I/Os on a HP 3000 Classic machine. The Classic retrieves its records block by block, not like the 2048 words per fetch on RISC HP 3000s. An elongation of 1 is ideal and attainable with a *SuperChained DetPack*. This is very good for a Classic system, but it won't necessarily deliver the most efficient results for a RISC system. (To help you, Adager's *SuperChained* option under a RISC machine automatically executes the *Chained* option.) Elongations of 2 but less than 3 may or may not be worth DetPacking. These should be double-checked with the "current I/O vs. best I/O" formulas to see if *DetPack* would be beneficial. (It's up to you to decide the ultimate benefit, since the time a *DetPack* ties up the database may sometimes outweigh the minimal performance gains attainable.) In general, if

you can cut your I/Os at least in half, DetPacking is worth the time.

Most of the time you will see the greatest performance gains using DetPack with elongations of 3 and greater. Through the regular usage of *HowMessy* and *Adager* you can keep the elongations to a minimum and thus improve the overall performance of IMAGE.

Additionally, if you regularly DetPack datasets, each maintenance session will take less time. Doing DetPacks on a regular basis means every DetPack can run up to 10 times faster than when elongation is high, due to the increased locality of the chain entries.

Through practice and understanding, you can develop a DetPack strategy solely on the value of "Elongation." For more information, read the *HowMessy* documentation in the *DOC* group of the Robelle account or in the *LIBDOC* group of Adager's Rego account.