IMAGE Performance (part 1)

Ken Paul
Adager Corporation
Sun Valley, Idaho 83353-3000 • *USA*
http://www.adager.com

At Adager we frequently get calls from database administrators (DBAs) who are experiencing performance problems with their IMAGE databases. Usually the DBA is convinced that increasing the capacity of some of the heavily used master datasets will solve the problem. To respond to these calls I have developed a checklist of questions that need to be answered by the DBA in order to pinpoint where the performance problem lies.

Adding or Retrieving?

The most important question the DBA must answer is whether the poor performance occurs when users are adding records to the database or when they are retrieving entries from the database.

Unfortunately, most DBAs do not know the software or the database well enough to answer this question. The DBA should talk to the third-party software supplier or the programmers to find out what certain programs are doing. The DBA should also talk to the end-users who are experiencing the problems so that they can tell the DBA what operation they are doing.

The correct answer to this first question can cut the research into finding the problem in half.

Adding Entries

I will now discuss the various reasons why adding entries to the database could be causing a performance problem.

Integer Keys

Integer keys *can* be a performance problem when adding records to a master dataset. Notice I did not say *are* a performance problem. Whether integer keys are a performance prob-

lem depends upon the data that is being assigned to the integer key. Because most people are aware of the problem with integer keys I would say that less than 10% of all master datasets use integer keys. They can usually be found in third-party software that has been around a while.

Clustering

When integer keys are used a concept known as "clustering" tends to take place within the master dataset. Because the IMAGE hash algorithm for integer keys "modulos" them, the records are usually placed one after the other forming a "cluster." Clusters are not inherently bad. It is possible to have a master dataset with an integer key and one large cluster but if that dataset has no secondaries there is no performance problem. An example of this would be a master dataset with a capacity of 10,000 containing 9,000 entries with key values 1 - 9,000. This dataset would have one very large cluster but it would not have any secondaries and performance would not be a problem at all!

Clustering becomes a problem when more than one cluster exists within the master dataset and one cluster begins to overlap another cluster. This usually occurs when large integer key values are used, such as 9 digit numbers which have the year as the first part of the number. Because of this numbering scheme new clusters are started within the master dataset every year. Things can be going along fine, and then your system can appear as if it hit a wall. This usually occurs when one cluster overlaps another. In order to add a new entry, IMAGE must scan for the next available space but it must read through the cluster before it finds an empty space. Every new record which is added must do the same thing.

To find out if you have a clustering problem:

1. Check the key type of the master. If it's an I or a J type your are more likely to have the clusters described above.
2. Use a tool such as DBLOADNG from the CSL or HOWMESSY from Robelle (which we include in your Adager tape as a courtesy from Robelle) to see what MAXBLOCKS and the percentage of secondaries are.
3. If MAXBLOCKS is high and the percentage of secondaries is non-zero you could have a clustering problem.

If you have a clustering problem, simply increasing the capacity to an arbitrary value is not going to guarantee a solution to the problem because you may still have overlapping clusters. You must find out what values are being placed in the key field and how those numbers are generated.

Using the key values that currently exist and projecting what will be added in the future, Adager Support can help you determine a capacity which separates your clusters.

Sorted Paths

Sorted paths are another area which MAY cause a performance problem when adding entries to a detail dataset. Notice again that I said MAY and not DO cause a performance problem. Sorted paths, like integer keys, have had a lot of bad press. Just like integer keys, there are good ("harmless") sorted paths and very bad ("sordid") sorted paths.

To decide if sorted paths are the cause of your performance problem first see if any sort fields are defined by doing an Adager Report Path of the detail dataset. If there are no sort fields, then that detail dataset is not the cause of your problem. If there are sort fields, you need to find out what data is being stored in that field.

A harmless path is one that causes very little overhead when adding a new entry. The most common use of a harmless sorted path is when a date field is used as the sort field. Usually the date field will contain today's date and so records are always placed at the end of the chain. The sort field may be used to insure that the chains are always in the correct order especially if someone adds an entry which was overlooked from last month and it needs to be added with that date.

If the data that is being placed into a sort field is random and the chains on that path tend to be long, then you are looking at a potential performance problem due to a sorted path. If the data is random, but the chains tend to be short, I doubt that this sorted path is causing your performance problem. Again use a tool like DBLOADNG or HOWMESSY to determine Max Chain Length and Average Chain Length for each sorted path.

Full Masters

It is true that as master datasets become more full, adding records to the dataset may take longer. This is usually (with the exception of integer keys) a very gradual degradation. The 80% rule for masters is a very good general rule but it is not the be all and end all. As databases have gotten larger and larger over the years the 80% rule may prevent a performance problem but it also may cause a disc space shortage problem.

Take, for instance, a master with a capacity of 5 million. You may be saying to yourself "Yeah. Right! Who has a master that big?". Believe me they exist and if these masters are kept less than 80% full over 1 million entries will always be available. 20% of 1000 is no big deal but 20% of 5 million can be a real

big deal and a lot of wasted disc space. Rules are nice but they should be guidelines and not strict dogma.

Many people have also stated that having more than 30% secondaries is unacceptable and capacities should be increased to lower this number. Once again, I have seen datasets with 50% secondaries and no performance problems. The key things to look for in masters is the Max Synonym Chain Length and the Average Synonym Chain Length as well as the percentage of secondaries. If I have 50% secondaries but 2 for a Max Chain Length and 2 for an Average Chain Length, I do not have a serious performance problem.

Next Time: Part 2

Next time I will talk about the reasons for performance problems when retrieving entries from an IMAGE database.