Understanding Prefetch in IMAGE

Ken Paul

Adager Corporation
Sun Valley, Idaho 83353-3000 • *USA*

www.adager.com

Randy Smith posted the following on the HP3000-L Internet list/newsgroup:

Would someone explain TurboImage Prefetch? I was reading Robelle's "What's Up DOCumentation", which talked about increasing TurboIMAGE performance by using prefetch.

- I can infer many things from the term prefetch as applied to a database, but what does it mean exactly for IMAGE?
- What are the benefits?
- What are the drawbacks?
- Are there any gotchas?

Background.

Prefetch is a flag that can be *enabled* for your database via DBUTIL. It was introduced in TurboIMAGE version C.03.13, which corresponds to MPE/iX version 3.1. By default, this flag is *disabled* for your TurboIMAGE databases and must be explicitly *enabled* using DBUTIL.

The term "prefetch" can have many different meanings. Usually, prefetch refers to file I/O, where the system is going out to disc to get more information before you ask for it. This type of prefetch can greatly enhance the performance of your programs, especially if the data that is being prefetched by the system is indeed the data your application needs. This usually is very beneficial with a *single* process that is doing serial reads of a file or dataset (this is typical in batch mode, for instance).

Unfortunately, this is not what is meant by the TurboIMAGE prefetch flag.

TurboIMAGE is currently *single threaded* when it comes to doing DBPUTs, DBDELETEs and, most recently, Critical Item DBUPDATEs. This means that only *one* process is actually add-

ing or deleting (or updating a critical item) at a time. In order to accomplish this, TurboIMAGE uses a semaphore lock known as the PUT/DELETE semaphore. This lock is *applied* before the updating of several blocks within the database and *released* at the end of the updating, to ensure data integrity.

The prefetch flag tells TurboIMAGE when to apply this semaphore lock.

- Before the existence of the prefetch flag (and with prefetch disabled), TurboIMAGE locks the PUT/DELETE semaphore before reading, updating, and writing every data block which is modified by the transaction. After this is done the PUT/DELETE semaphore is unlocked.
- With prefetch enabled, TurboIMAGE reads all the data blocks that are needed for the complete transaction before the PUT/DELETE semaphore is locked. Once the semaphore is locked, TurboIMAGE updates and writes the data blocks which are modified by the transaction before unlocking the semaphore.

It is the holding of the semaphore lock for a shorter period of time which could improve the throughput of data on your system, but this is not guaranteed. There is the possibility that after the data is prefetched—and before the semaphore lock is applied—another process may flush it out of memory before TurboIMAGE is able to update it. If this is the case, TurboIMAGE must go out to disc to re-read the block and this extra I/O activity will have a negative impact on your database performance.

Effectiveness of prefetching

The Prefetch flag has always been a "your mileage may vary" flag since its introduction. Certain conditions need to exist on your system in order to—possibly—gain benefit from turning on this flag.

First, the system should have "adequate" (a wonderfully nebulous word) memory available to handle the increased data page locality, as well as adequate processor capability to handle increased concurrency of processes.

Second, applications should make numerous calls to DBPUT, DBDELETE (or, most recently, DBUPDATE to critical items).

Third, multiple processes must be actively updating a database before a benefit can be realized.

The HP Database Lab has said that they have seen 5% to 10% improvements in the number of transactions that could be

processed in benchmarks as a result of enabling prefetch. They also mention that it helps in situations where you have excess CPU time and you are running update-intensive applications (I think they mean intensive in terms of DBPUT/DBDELETE/ Critical-item DBUPDATE).

The 35% performance improvement that is mentioned in Robelle's "What's Up DOCumentation" is news to me. I have never seen or heard of such a high value. I consider the prefetch flag a "try it, you might like it" flag.

Problems with prefetch

There was a problem with prefetch under TurboIMAGE versions C.04.06 through C.04.08, when DBPUTs to a detail dataset were failing because this flag was set and there were some uninitialized variables which caused problems. This bug, however, did *not* cause database corruption. The fix was to disable the database for prefetch until the next version of TurboIMAGE was released via a patch.

I also ran into a site which was experiencing bizarre "DBG disabled" messages while opening a database. The messages could not be traced back to broken chains. After analyzing the database and finding no broken chains, we noticed that prefetch was enabled and, upon disabling it, the problem disappeared. This was on TurboIMAGE version C.04.19.

My advice

Enable the prefetch flag and see if your environment improves noticeably. You can always turn prefetch off quite easily if any problems appear.

Gratitude

I would like to thank Marguerite Bryan of the HP Response Center and Jim Sartain, formerly of the HP Database Lab, for information which I used in this article.