

---

# *The Fourth Bear of IMAGE*

*Fred White*  
*Senior Research Scientist*

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

---

***Introduction*** The location of users, programs and databases relative to each other on the nodes of an HP3000 network can be critical to program and network performance.

When a user is remote from the program, additional memory and CPU resources are consumed at both nodes to support user I/O.

When a database is remote from the program, additional memory and CPU resources are consumed at both nodes to support database access.

Program performance is also negatively impacted by the presence of a remote user and/or a remote database.

Some of the questions we will attempt to answer are:

1. Is remote user access significantly slower than local user access?
2. Is remote database access significantly slower than local database access?
3. If the user and database are on different nodes is it better to run the program on the user's node or on the database's node?

All of these questions relate to the overhead associated with access to a remote user (or remote database) arising from the use of HP3000 network facilities (DS or NS) whose use requires each node to provide resources (in the form of memory, virtual memory, tables and CPU) that would otherwise not be needed.

***Transactions*** Each transaction of a database application begins with the program prompting the user with a terminal write and then accepting data with a terminal read. There may be many such

prompt/response pairs after which the program performs database access determined by the application and the user's responses to the prompts.

This process is repeated until the user input causes the program to perform a database close followed by program termination.

We are not interested in the performance of the program itself so the rest of this paper takes this performance as a given. However, we are interested in how much the performance of such a program would be impacted by either remote database access and/or remote user access.

The term "remote database" is used here to indicate that the node of an HP3000 network on which the database is resident differs from the node on which the program is running. A similar definition applies to the term "remote user".

A remote user (or database) will be said to be "R-remote" from a program if R network transfers occur whenever the program writes to the remote user (or database). In this definition, R=0 implies that the user (or database) is local.

### ***1-REMOTE USER ACCESS***

All user access is initiated by the program.

The sequence for 1-remote user access is:

1. The program calls a file management intrinsic.
2. File management packages the call information and invokes the network facility (DS or NS).
3. The network facility transmits the package to the user's node.
4. The network facility at the user's node passes the package to the user's command interpreter process.
5. The command interpreter process performs the same file management call as in step 1 above.
6. File management performs the terminal I/O.
7. The command interpreter process packages the result and invokes the network facility.
8. The network facility transmits the package to the program's node.
9. The network facility on the program's node awakens the program.
10. File management returns the result to the program.

Note that, for local user access, only steps 1, 6 and 10 are performed so that steps 2 thru 5 and 7 thru 9 constitute the additional overhead of 1-remote user access.

For character mode prompting, this overhead consumes about 8 times as much CPU as that required for local user access with about half of this consumed on the program's node and half on the user's node.

At first glance, one might conclude that 1-remote user access would be about 8 times as slow as local user access. In point of fact, it is typically less than 5 percent slower.

To understand why, first note that in order for a program to acquire input from a user it must perform a prompt/response pair so that the user access sequence shown must be traversed twice. Once for writing and once for reading.

For each of these it is necessary that we approximate the average wall time consumed by steps 2 thru 5 and 7 thru 9. Our estimate of 120 milliseconds was derived from Performance News Notes, published by HP, reflecting some LAN/3000 and NS/3000 performance tests run stand-alone on two HP3000/Series 48s and two HP3000/Series 68s.

Proceeding on the assumption that this estimate is quite reasonable, we can conclude that the wall time remote user overhead for a typical prompt/response pair would be on the order of 240 milliseconds.

The combined wall times for performing the terminal I/O of step 6, including think time and data entry time, is typically measured in seconds.

If the combined prompt/response time of step 6 is 5 seconds, then the remote user access overhead of 240 milliseconds would decrease user access performance by about 5 percent.

I have not investigated the degree to which this overhead would impact remote user access when the prompt/response pair is performed in page mode. My guess is that the remote user access overhead would increase by a factor of ten or more but that the combined prompt/ response time would also increase by about the same factor so that similar conclusions would result.

### ***1-REMOTE DATA-BASE ACCESS***

All database access is initiated by the program.

The sequence for 1-remote database access is:

1. The program calls an IMAGE/3000 intrinsic.
2. IMAGE packages the call information and invokes the network facility (DS or NS).
3. The network facility transmits the package to the node of the database.

4. The network facility at the node of the database, passes the package to the user's command interpreter process.
5. The command interpreter process performs the same IMAGE intrinsic call as in step 1.
6. IMAGE processes this call. This may involve disc I/Os.
7. The command interpreter process packages the result and invokes the network facility.
8. The network facility transmits the package to the program's node.
9. The network facility on the program's node awakens the program.
10. IMAGE/3000 returns the result to the program.

Note that, for local database access, only steps 1, 6 and 10 are performed so that steps 2 thru 5 and 7 thru 9 constitute the additional overhead of remote database access.

If we denote the wall time of step  $i$  by  $T_i$ , then the performance of an intrinsic employing remote database access will be  $N$  times as slow as the same intrinsic using local database access where:

$$N = (T_1 + T_2 + \dots + T_{10}) / (T_1 + T_6 + T_{10})$$

As with remote user access, the wall time overhead of steps 2 thru 5 and 7 thru 9 is about 120 milliseconds so that we have:

$$N = (T + 120) / T = 1 + 120/T$$

where  $T$  is the elapsed time estimate, in milliseconds, of the intrinsic when performed locally. Although the impact on response time is the same for all IMAGE intrinsics, namely about 120 milliseconds, the relative impact varies with the intrinsic being called.

The DBINFO intrinsic, in some modes, is able to respond to the user locally, based on the contents of the remote database control block (RDBCB). In such cases, this equation does not apply and there is no remote database overhead.

In all other cases the intrinsic may perform an average of  $M$  disc I/Os where  $M$  is non-negative and may be a fraction.

If  $M=0$ ,  $T$  is generally between 4 and 8 milliseconds so that  $N$  will be between 16 and 31.

For  $M=1/2$ , half of the calls would involve no I/O with a  $T$  value of about 5 and the other half would require 1 I/O with a  $T$  value of about 30. The time to complete these two would be 35 seconds locally and 275 milliseconds remotely so that  $N$  would equal 275/35.

Values of  $N$  for various values of  $M$  are:

| M   | N     |
|-----|-------|
| 0   | ~20   |
| 0.5 | ~8    |
| 1   | 5     |
| 2   | 3     |
| 4   | 2     |
| 8   | 1.5   |
| 16  | 1.25  |
| 32  | 1.125 |

Thus, database access for a given transaction will be at least twice as slow for remote access as for local access as long as the average number of disc I/Os of the IMAGE intrinsic (excluding those DBINFO calls which are responded to locally) involved in the transaction is less than 5.

**REMOTE USER versus REMOTE DATABASE**

So far, we have seen that 1-remote user access is generally less than 5 percent slower than local user access and that 1-remote database access may very easily be 5 or 10 times as slow as local database access.

We now ask the question: "In a 2-node network, with a user logged on node 1 and a database resident on node 2, is it better for a program which accesses the database to be run on node 1 or on node 2?"

The knee-jerk answer is "node 2" so that the program avoids the performance degradation of remote database access at the expense of the performance degradation of remote user access.

Fortunately, in the vast majority of cases (99%?), this is the correct answer.

To see this, let  $X$  denote the number of prompt/response pairs for the average transaction and  $Y$  denote the number of IMAGE intrinsic calls per average transaction.

The question becomes: "Is it better to expend  $240X$  milliseconds in support of remote user access or  $120Y$  milliseconds in support of remote database access?"

From this we see that remote user access will be better than remote database access whenever  $240X$  is less than  $120Y$ .

Our conclusion is then valid for those cases in which  $Y$  is greater than  $2X$ .

It happens that, for an  $R$ -remote user, the additional wall time per network transfer is approximately linear so that the wall time overhead for each prompt response pair is  $240R$  milliseconds.

The same linearity happens in remote database access yielding a wall time overhead of  $120R$  milliseconds for an R-remote database.

Because of this linearity, the question of locality for best performance when the user and database are on distinct nodes of any network has the same answer, regardless of the number of intermediate nodes.

**Summary** If, in spite of everything, you find it necessary to use remote database access, you can estimate transaction degradation time due to remote database access as follows:

1. Let  $M$  denote the number of IMAGE calls in the transaction (excluding DBINFO calls satisfiable locally).
2. Let  $N$  denote the distance, in nodes, between the database and the program ( $N=0$  for local database access).
3. Then, the transaction degradation time is approximately  $M \times N \times 120$  milliseconds.

As a final caveat, note that large internode distances, low baud rates, network contention and CPU contention at any node can all increase this degradation time.