
WHITE PAPER

AUGUST 1997

Database Gateway Use In Heterogeneous Environments

Table of Contents

Introduction	1
What is a Gateway?	1
Key Attributes Of A Gateway	1
Gateway Vendors for the HP 3000	3
HP ALLBASE/NET	3
Oracle Transparent Gateway for IMAGE/SQL	3
Information Builders EDA/SQL	4
Sybase Open Client/Open Server & Direct Connect Gateway	4
Oracle Transparent Gateway Architecture	5
Process On The Server and Client Nodes	6
Setup Steps	9
New Features Offered by Oracle	
Transparent Gateway Release 4.0	12
Transactional Integrity: Two-Phase Commit	12
Flexibility & Improved Functionality: Passthrough Features	13
Performance Tuning Tips	14
CASE 1: Use DBOPEN Modes Efficiently To Minimize Locking	14
CASE 2: TurboIMAGE Rollback Limitation	15
CASE 3: Client/Server Gateway Connection	15
CASE 4: Third-Party Indexing Tools	15
Summary	16
Appendix A	17
Listing 1: Parameter Values	17
Listing 2: Sample Configuration Files	18

Introduction

The traditional HP 3000 based application often uses the TurboIMAGE database management system as a data source. As traditional applications grow or new applications are added, either modifications are made to an existing TurboIMAGE database or new databases are added to support the changes. Regardless, the data remained homogeneous and access uncomplicated.

Today, when business needs require application modification, replacement or the addition of new applications, the data source for these applications often is a relational database management system residing on either the HP 3000 or another platform. These new applications frequently require access to existing HP 3000 data sources, the TurboIMAGE databases. Until recently, that access was accomplished through user application programming. Now, access can be provided by database gateways from various vendors such as Hewlett-Packard, Oracle, Information Builders and Sybase.

This paper examines the characteristics of database gateways to HP 3000 data sources. Using the Oracle Transparent Gateway for IMAGE/SQL as a primary model, the paper addresses gateway architecture, gateway setup/usage and performance tuning tips.

What is a Gateway?

Gateways are powerful coexistence tools that facilitate enterprise-wide information management. They provide access to heterogeneous data sources residing on heterogeneous platforms at distributed locations. Gateways integrate various data sources and create the appearance that all data resides in a single, logical relational database.

Key Attributes Of A Gateway

A key attribute of a gateway is the ability to provide many levels of transparency. In addition a gateway provides: operating system transparency such that users do not need to know the operating system on which the data resides; geographic transparency in that end-users only need to know the name of the relational table(s) to be accessed (they do not need to know the physical location of the data); network transparency, allowing users to access the data regardless of the network protocol(s); and finally, it provides data storage transparency that permits users to access data regardless of the data storage methodology used.

These levels of transparency can definitely enhance end-users' ability to access and analyze data. For example, in a single request, a user can integrate order information from Oracle7 on an HP 9000 with manufacturing parts information from a TurboIMAGE database on an HP 3000 without having to know where each piece of information is located, how it is stored or what operating system each server uses. Transparent data access can protect a company's investment in data, applications and hardware. Furthermore, it can extend a user's data management power across heterogeneous platforms.

Another key attribute of a gateway is to provide a turnkey solution for end-users. Instead of requiring users to do extensive programming and customization in-house, each gateway product enables complete and ready-to-use solutions for which no programming is needed. Once the gateway product is configured to access data from various supported sources, the user can begin integrating and managing data immediately.

Other common gateway attributes are as follows:

- Gateways provide read/write data access and enable ad-hoc queries to heterogeneous databases using Structured Query Language (SQL).
- Gateways provide easy data migration where data moving from one source to another data source can be accomplished with SQL statements.
- They allow users to leverage applications from their existing environment by allowing connectivity between HP 3000 databases and other databases on other platforms. The ability to interoperate and exchange data is critical to the computing success of a business.
- They provide security based on a combination of specified database servers and native operating system mechanisms.
- Gateways use client/server architecture, enabling the user to select the most cost-effective platform to serve as the client, and to use in-house or vendors' tools and applications to access the desired platform as the server.

Gateway Vendors for the HP 3000

Multiple vendors provide gateway products that can interface with the HP 3000 platform and other UNIX-based platforms. Some of these vendors are Hewlett-Packard, Oracle, Information Builders and Sybase. Below are brief descriptions of their products.

HP ALLBASE/NET

ALLBASE/NET from HP is a data access product that provides transparent, remote access to TurboIMAGE and ALLBASE/SQL data as though the data were stored locally. ALLBASE/NET uses a client/server architecture. The application runs on either an HP 3000 or HP 9000 client; the server can be either an HP 3000 running TurboIMAGE or ALLBASE/SQL, or an HP 9000 running ALLBASE/SQL. ALLBASE/NET uses TCP/IP for network access.

ALLBASE/NET is bundled with ALLBASE/SQL and IMAGE/SQL on both HP 3000 and HP 9000 platforms. It allows queries or updates to TurboIMAGE and ALLBASE/SQL databases anywhere on the network using SQL calls. This solution provides economical and attractive methods to sharing data across HP platforms.

With ALLBASE/NET, users can create an alias file using SQLUTIL to define a remote database that will appear as a local database. From then on, the user can simply access and manage the remote database as a local database.

Oracle Transparent Gateway For IMAGE/SQL

Transparent Gateway for IMAGE/SQL, Oracle's second-best selling gateway, provides Oracle database users from more than 90 platforms with read and write access to TurboIMAGE data. By providing transparent access, TurboIMAGE data appears to be part of an Oracle database.

With the Oracle Transparent Gateway for IMAGE/SQL, users may be unaware that the data they are accessing resides as TurboIMAGE data on an HP 3000. This gateway provides data access via SQL to TurboIMAGE data that has been defined, mapped, and attached to IMAGE/SQL's database environment (DBE). With this gateway, TurboIMAGE's underlying network data model appears relational to client applications.

The Oracle7 Server functions as an integrator by managing communications to various servers and the gateway for IMAGE/SQL. It provides services such as distributed optimization, distributed joins, and transaction coordination between Oracle and a TurboIMAGE data source. The gateway supports the TCP/IP protocol, and Oracle SQL*NET provides connectivity to the HP 3000. The SQL*NET product is required on all accessing and accessed network nodes.

Information Builders EDA/SQL

Information Builders, Inc. (IBI) also provides a database interoperability product called EDA/SQL (Enterprise Data Access). EDA/SQL allows direct data access transparently to heterogeneous database management systems on heterogeneous platforms. On the HP 3000, it allows TurboIMAGE, ALLBASE/SQL, ORACLE and KSAM data to be retrieved from a multi-platform and multi-database environment. EDA/SQL uses TCP/IP or NETIPC for network access. In addition, EDA/SQL allows HP 3000 users to access data on many other platforms. IBI also provides a window-based desktop decision support and analysis tool called FOCUS, which works in conjunction with EDA/SQL.

Sybase Open Client/Open Server & Direct Connect Gateway

Bradmark and Sybase are providing several Sybase gateway products for the HP 3000 platform. Open Client/Open Server, a set of APIs, is now available. HP 3000 customers can now programmatically access Sybase databases residing on other platforms from an HP 3000 environment using these APIs. Alternatively, Image and Allbase on HP 3000 can be accessed programmatically from a Sybase environment using these APIs. Later this year, Bradmark will release DirectCONNECT. This product will allow transparent access, as opposed to programmatic access, to Image and Allbase databases from a Sybase environment.

Figure 1 shows various ways of accessing data among HP 3000, HP 9000 and other platforms. Database connectivity can be accomplished with multiple gateway products. The ability to coexist among heterogeneous platforms provides MIS with the flexibility to leverage each platform's strength.

Figure 1.

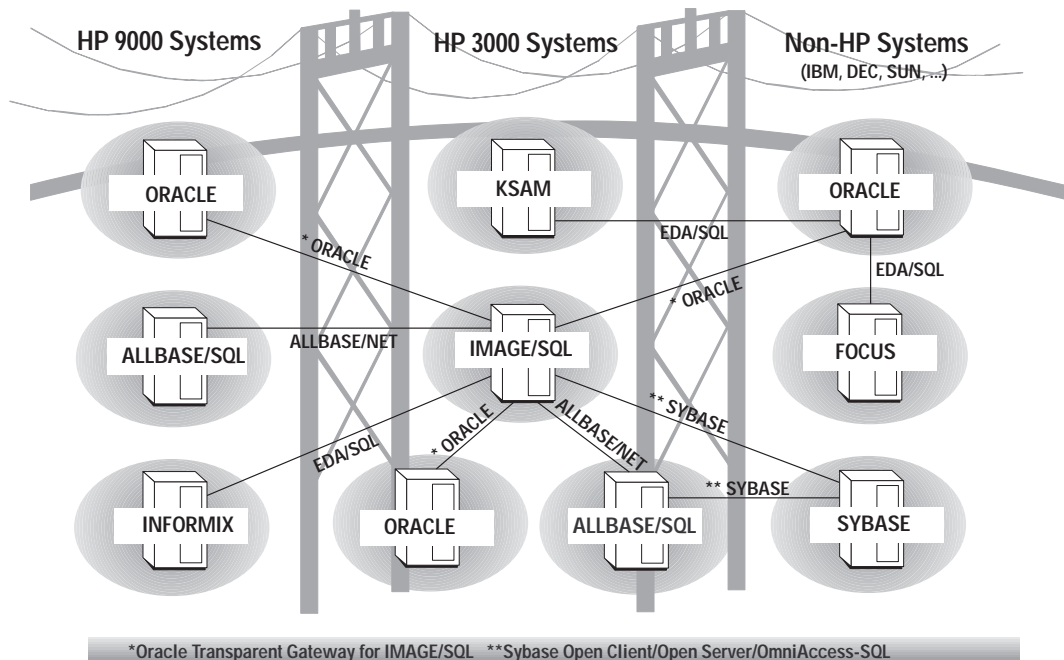


Table 1. Gateway Products Information

HP ALLBASE/NET	Kriss Rant, Hewlett-Packard	(408) 447-6653
Oracle Transparent Gateway for IMAGE/SQL	Jennie Hou, Hewlett-Packard	(408) 447-5971
IBI EDA/SQL	Marlene Nesson, IBI	(212) 736-6250
Sybase Open Client/ Open Server/ Direct Connect	Mike Whitely, Bradmark	(800) 275-2723

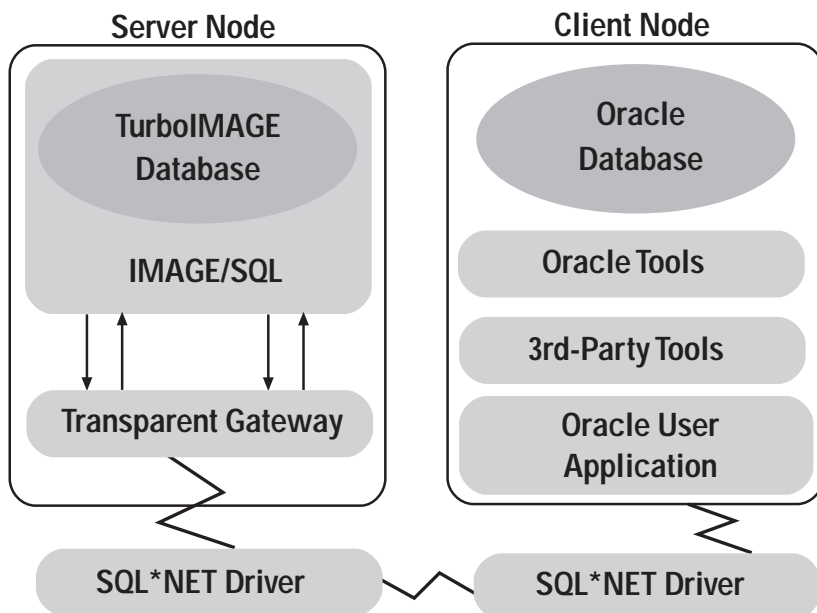
Oracle Transparent Gateway Architecture

The Oracle Transparent Gateway for IMAGE/SQL is the model used for discussing architecture, setup/usage and performance tuning.

The Oracle Gateway for IMAGE/SQL emulates remote Oracle7 servers, allowing users to integrate TurboIMAGE databases into an Oracle7 environment. The gateway resides on the server node where the TurboIMAGE data source also resides. It uses IMAGE/SQL's mapping of TurboIMAGE data into relational table formats. The mapped data can be accessed from the client node with:

- An Oracle integrating server
- An Oracle tool (e.g., Oracle Forms)
- A user application or
- A third-party tool that utilizes Oracle's open interface standards

Figure 2. The architecture layer



Processes On The Server and Client Nodes

Several processes need to be done on the server and client nodes when setting up and using the gateway and SQL*NET V2. Below are the major steps:

Server

1. Start SQL*NET V2 on the server node.

To invoke SQL*NET V2, the user must set up the TNS_ADMIN environment variable to point to where the *listener.ora* SQL*NET V2 configuration file resides and start the listener process. The *listener.ora* file contains information such as the port number, protocol, hostname and etc.

The SQL*NET V2 utility LSNRCTL creates a child process TNSLSNR and passes all parameters to that child. TNSLSNR then adopts itself to PROGEN, and consequently becomes a system process.

Now, the TNSLSNR process is waiting for connection requests to come. It listens on the port specified in the *listener.ora* configuration file.

Server

2. Start the gateway server process on the server node.

The GTWCTL R command executed from the Oracle software account streams a gateway process job. The gateway server process GTWSRV is up and running at this point.

Client

3. Set up the TNS_ADMIN environment variable on the client node.

This variable must be set so the SQL*NET V2 knows where the *tnsnames.ora* file resides. In this file, data about the server such as hostname, port number and protocol are provided, as well as an alias to be used to connect strings in database links.

Client

4. Client node fires off a request.

For example, the client issues the following command:

```
sql> connect system/manager@SQLALIAS
```

where SQLALIAS is the connect string name in the *tnsnames.ora* file.

From the SQLALIAS, the client should be able to find the server hostname, port ID and protocol. The client sends a "connect" request to the server via the port specified in the *tnsnames.ora* file.

Server

5. Server node processes the client request.

The TNSLSNR process on the server node receives the client “connect” request from step 4 above. The server does the following:

- Gets the “connect” request through specified port.
- Passes the request and determines if the request is for the gateway or the Oracle kernel.
- Creates a child ORACLE if the request is for Oracle, or creates a child GTWEX if the request is for the gateway.
- Passes connect information to the child, including a new port Id for the client.
- If the child is ORACLE, it remains as the child of the TNSLSNR process; if the child is GTWEX, it adopts itself to the GTWSRV process (from step 2 above).
- The child sends a message to the client with the new port information.

Client

6. Client node request processed.

If the request from step 5 above is for the gateway, at this point, the client is “connected” and has the port to GTWEX.

The GTWEX gets data from the *init<GATEWAY_SID>.ora* configuration file, which resides in the */<ORACCT>/GTW/admin* directory. This file provides the IMAGE/SQL DBE and GATEWAY_SID names. The GTWEX is mapped to the IMAGE/SQL DBE, which in turn points to the physical TurboIMAGE database. At this point, the gateway to TurboIMAGE database is set.

Client

7. Set up the database links on the client node.

On the client node, using Oracle SQLDBA or SQLPLUS the database link to TurboIMAGE database is set up. The database link name is unique.

Example:

```
SQLDBA> connect system/manager;
QLDBA> create database link gtwlink
        connect to "mgr.tiacct"
        identified by "jack/jill"
        using 'bucketalias';
```

Client

8. Access TurboIMAGE data from the client node.

At this point, the client knows how to access the TurboIMAGE data from the Oracle database. Now, to issue an SQL statement:

```
sql> select * from <TurboImageDB>.<Table>@<linkname> ;
```

For example:

```
sql> select * from TIDB.EMPLOYEES@GTWLINK ;
```

Through the above select statement, the client can resolve user/passwords, IMAGE/SQL mapping and the physical TurboIMAGE database. The client can carry out the required task.

Client/Server

9. Client/Server Connection.

The client connection to the server node is maintained until the client database user disconnects. At this point, the client and server are ready to process more requests.

Setup Steps

The major process steps described in the previous section can be used to set up a gateway between a TurboIMAGE database and an Oracle database on an HP 3000 system.

1. Set up the SQL*NET V2 and Gateway configuration files.

On the server node, the required SQL*NET V2 configuration files are *listener.ora* and *sqlnet.ora*. On the client node, the required SQL*NET V2 configuration file is *tnsname.ora*.

For this example, both the server and client nodes reside on the same hardware system, so all above files will reside in the same Oracle software account <ORAACCT>. The default directory containing these files is <ORAACCT>/NETWORK/admin. However, these files can be placed anywhere on the system as long as the "TNS_ADMIN" environment variable is set accordingly.

On the server node, set up the gateway initialization file, *init<gateway_sid>.ora*. This file *always* resides in the <ORAACCT>/GTW/admin directory.

The key things to be aware of when configuring these files are:

- The port number must be identical in the *listener.ora* file and the *tnsnames.ora* file. The host names must be set properly in these files.

In the *listener.ora* file,

- The gateway_sid name must match the db_name in the gateway initialization file.
- One SID_NAME entry must be the TurboIMAGE database name.
- The ORACLE_HOME must be set to the user's Oracle software account.
- The PROGRAM must be set to "GTWEX".

In the *tnsname.ora* file,

- The SID name must be the TurboIMAGE database.
- The connect string name or the alias name can be anything; it doesn't have to be identical as the gateway_sid name in the *listener.ora* file.

In Appendix A *Listing 1* contains parameters and their values. They are used in the examples throughout this section. *Listing 2* contains sample *.ora files.

2. Set up the TNS_ADMIN environment variable.

The TNS_ADMIN environment variable points to the location of the *.ora file configured in step 1 above.

CI: setvar tns_admin “/<ORAACCT>/NETWORK/admin”

Example: setvar tns_admin “/ORACLE7/NETWORK/admin”

3. Start the SQL*NET V2 listener.

CI: lsnrctl.pub.<ORAACCT> start

Example: lsnrctl.pub.oracle7 start

4. Start the gateway server process on the server node in the user's Oracle software account.

CI: gtwctl.pub.<ORAACCT> r

Example: gtwctl.pub.oracle7 r

5. Mapping the TurboIMAGE database to IMAGE/SQL on the server node.

CI: imagesql.pub.sys

>> set turbodb <TurboIMAGE database name>
(Example: set turbodb tidb)

>> set sqldb <dbe name>
(Example: set sqldb tidbe)
(Create a DBE (database environment)? Y/N: Y)

>> attach
(Attach the database to DBE)

>> display map
(Lists the attached database tables and mapped database fields with original and mapped datatypes.)

>> display users
(Lists the user information and TurboIMAGE dbopen mode.)

>> exit;

6. Verify that the IMAGE/SQL mapping is done successfully by going into ALLBASE/SQL's isql on the server node.

CI: isql.pub.sys

```
isql=> start dbe '<dbe name>';
      (Example: start dbe 'tidbe';)
```

```
isql=> select * from <ti_database name>.<table_name>;
      (Example: select * from tidb.employees;)
      ISQL should display the specified table with complete
      rows of data.
```

```
isql=> exit;
```

7. Create the Oracle database link through the gateway on the client node.

CI: sqldba lmode=y

```
SQLDBA> connect system/manager ;
SQLDBA> create database link <gtw_link_name>
      connect to "<USER>.<TI_ACCT>"
      identified by "<userpass>/<acctpass>"
      using '<connect-string/alias>';
Statement processed.
```

where <TI_ACCT> is the user's account containing the TurboIMAGE database.

where <userpass> and <acctpass> are passwords for the user logon to <TI_ACCT>.

where <connect-string/alias> is a dummy alias name the user chooses in the tnsnames.ora file.

Example:

```
SQLDBA> connect system/manager;
SQLDBA> create database link gtwlink
      connect to "mgr.tiacct"
      identified by "jack/jill"
      using 'bucketalias';
```

8. Establish TurboIMAGE data access through Oracle

At this point, the user is set up for reading/writing TurboIMAGE data through Oracle's gateway. To verify, enter a simple select statement specifying a table linked through the gateway:

```
SQLDBA> select * from
      <ti_database_name>.<table>@<gtw_link_name>;
```

Example: SQLDBA> select * from tidb.employees@gtwlink ;

New Features Offered by Oracle Transparent Gateway Release 4.0

Gateway to Image/SQL, Version 4.0, released in February 1997, is certified for MPE/iX 5.0 and 5.5. Transparent Gateway to Image/SQL 4.0 requires the SQL*Net listener from RDBMS 7.2.3. Transparent Gateway 4.0 has additional features such as passthrough to control Image/SQL locking strategies and two-phase commit. The integrating server can be any version or Oracle7, but if the Oracle database resides on the HP 3000, using RDBMS 7.2.3 is recommended to avoid SQL*Net complications.

Transactional Integrity: Two-Phase Commit

An important new feature of release 4.0 is its powerful two-phase commit that extends distributed update capabilities and transaction-integrity protection to all heterogeneous data stores, such as users could protect updates across multiple Oracle databases. The two-phase commit ensures that changes requested in a transaction are either all stored and permanently committed in each data source, or that all are rolled back to their original state.

In the first phase, the Oracle database prepares to commit the change to the desired data. Oracle then tells the IMAGE source to commit. The second phase is an Oracle database check to verify whether the IMAGE source has successfully committed the change. If it has, then the Oracle database permanently commits its own change. If the changes have not been successfully committed, Oracle rolls everything back to the original status. All data sources remain in sync, and the integrity of the data is maintained.

For example, consider a banking transaction transferring funds from a savings account to checking. The debit table on an IMAGE/SQL data source is reached via Oracle Transparent Gateway for IMAGE/SQL; the credit table is on an Oracle7 server (also known as the integrating server.) An end-user can issue the following statements from a SQLPLUS session to insert data into both tables:

```
SQLPLUS>Insert to Debit@savings values (500);
```

```
SQLPLUS>Insert to Credit@checking values (500);
```

```
SQLPLUS>Commit;
```

The "Commit" statement consigns both the foreign database (IMAGE/SQL) and the integrating server database. In case of failure, all sites participating in the distributed update transaction will roll back, thus maintaining the integrity of the databases involved in the transactions.

This is particularly important for enterprises with mission-critical data stored on HP 3000 systems. Access to this data is now possible, and completely transparent to the user, eliminating the need to migrate data to another system.

Flexibility & Improved Functionality: Passthrough Features

Oracle Transparent Gateway for IMAGE/SQL Release 4.0 uses the passthrough feature to control session-level locking from within a transaction. Users can issue a command to control IMAGE/SQL locking within a Gateway session without aborting the session, resetting locking level, or having to reinitiate the session with the new set-up level of locking.

In previous versions, locking could only be controlled globally, which meant one locking mode for all transactions performed in a gateway session. Altering the locking level would then require aborting the current gateway session to make the change effective. With the Passthrough feature, this can be done easily via a SQLPLUS statement. An example of such a SQL statement follows:

```
SQLPLUS>execute  
gtw_sql.gtwpass@gtwlink('SET SESSION ISOLATION LEVEL CS; ',  
'GTW');
```

where gtwpass is the statically linked-in procedure for Oracle Transparent Gateway for IMAGE/SQL; gtwlink is the Oracle database link to access the IMAGE data; and 'GTW' represents the Oracle Gateway to the foreign data source (IMAGE/SQL).

The function of the above SQL statement is to pass a parameter to the procedure gtwpass, which then sets the session level isolation to CS (Cursor Stability) from within a gateway session.

This capability provides greater connectivity between HP 3000 databases, Oracle databases and a variety of other databases – allowing interoperability, exchange and integration of information from within any of the databases. The end result is overall improved productivity, and more effective utilization of IMAGE/SQL data resources.

Performance Tuning Tips

Gateway performance can be improved in several ways. The first case addresses the efficient use of the TurboIMAGE DBOPEN modes. The second case addresses the TurboIMAGE rollback limitation and how best to work with it. The third case addresses the client/server gateway connection. The fourth case addresses usage of third-party indexing tools on TurboIMAGE.

CASE 1: Use DBOPEN Modes Efficiently To Minimize Locking

The gateway, like all TurboIMAGE applications, will lock out other users from either some rows or entire datasets (tables) when using DBOPEN MODE 1-4, (READ/WRITE), even when doing reading only. Locks can then be released if a rollback or commit is issued. Locking occurs because the gateway product is designed to use the default transaction mode “RR” (Repeatable Reads) from IMAGE/SQL when the DBOPEN mode is set to READ/WRITE. The “RR” mode ensures that the data selected will be identical through various repeated reads; which means that data needs to be locked to ensure consistency. However, in IMAGE/SQL, users can override the default “RR” mode by specifying other modes at the beginning of a transaction. This option to choose other transaction modes is now implemented in Release 4.0 via the passthrough feature.

Several approaches are suggested to avoid unnecessary locking:

- For greatest flexibility, use Release 4.0's Passthrough feature to provide transaction-level locking.
- If only the READ capability to the TurboIMAGE database is needed, and “RR” is not, then set the DBOPEN modes to READ ONLY, modes 5-8. (Be sure that the modes set are compatible with existing applications.) With READ ONLY, no locking will occur.
- If both READ ONLY and READ/WRITE capabilities are needed, users can minimize the “locking” hits by creating two users for the TurboIMAGE database account. Set up one user with the READ ONLY capability and the other user with the READ/WRITE capability. For each of these users, define a unique database link through Oracle SQLDBA. By doing this, the gateway will not obtain locks for the READ ONLY user, who can have unlimited access to TurboIMAGE data via a gateway as long as the same data is not being accessed by a READ/WRITE user. If there is only a READ/WRITE user instead of two users with different capabilities, locking will occur even during reading. Thus, the locking hits would be much higher.
- If the DBOPEN mode is set to READ/WRITE, users should make sure that each transaction is followed with a commit. A commit or a rollback will unlock the data for access by others. This way, the locks will be released after each transaction.

CASE 2: TurboIMAGE Rollback Limitation

In IMAGE/SQL, the rollback capability is supported by enclosing every transaction with a pair of DBXBEGIN and DBXEND calls. By doing this, the user can roll back a transaction if needed. However, the problem with the rollback capability is the limit placed on the rollback log buffer size.

The transaction manager on an HP 3000 limits the buffer size to 4MB, so if a user wants to commit a transaction which is greater than 4MB, the transaction cannot be completed. In TurboIMAGE, the user can override the rollback capability by not enclosing a transaction with a pair of DBXBEGIN and DBXEND calls; this is not the case for gateway access through IMAGE/SQL. This limitation is passed on to Oracle through the gateway to IMAGE/SQL to TurboIMAGE. To work around this limitation, it is best to divide the workload into units smaller than 4MB. This way, the limitation will not occur.

Another workaround is to write stored procedures (using Oracle PL/SQL) to handle the size of the workload. By using stored procedures, the user can manipulate the number of rows, and thus the size of the workload.

CASE 3: Client/Server Gateway Connection

When the gateway and the SQL*NET V2 processes have been set up, the connection between the server node and the client node is maintained until the user disconnects from the database. As long as the initial connection has been made, subsequent requests from the client to the server can be processed very quickly. The trick is to stay connected as long as possible, rather than connecting and disconnecting repeatedly.

In addition, two other tuning tips used for SQL*NET V2 can also be applied when using the gateway. First, for a long hostname file (e.g., HOSTS.NET.SYS,/etc/hosts), it is advantageous to put the server node and the client node on the top of the list. This minimizes the lookup time. Second, do not turn on the tracing in SQL*NET V2. Tracing is costly in terms of overhead. Except when debugging, tracing is normally turned off. The parameter "TRACE_LEVEL_LISTENER" can be found in the LISTENER.ORA file.

CASE 4: Third-Party Indexing Tools

Several third-party indexing tools are available for the TurboIMAGE database. The Oracle Transparent Gateway for IMAGE/SQL can transparently support any of these tools that IMAGE/SQL can support. (TurboIMAGE has been enhanced to support third-party indexing since 1996.) Use of these third-party indexing tools can improve the search performance.

Summary

Using database gateways to access heterogeneous data sources residing on heterogeneous platforms can enhance a user's data management power and protect a company's investment in data, applications and hardware. Several vendors can provide this technology to HP 3000 users. Although these gateways may vary architecturally, they share some common attributes such as providing many levels of transparency, providing a turnkey solution for end-users, using client/server architecture and allowing users to leverage their existing applications.

As seen with the Oracle Transparent Gateway for IMAGE/SQL example, setting up a gateway involves a series of steps on both the server and client nodes. Once the gateway is configured, TurboIMAGE data on an HP 3000 can be integrated into an industry-standard relational database such as Oracle, and can be accessed via SQL commands.

As for tuning, the overall gateway performance can be tuned, but users need to be aware of the limitations set by the gateway used as well as those of IMAGE/SQL.

The ability to interoperate and exchange data among heterogeneous platforms is crucial to a business' success and to ensuring the long-term viability and value of a company's computing investment. Gateway technology can definitely meet this pressing demand for coexistence in today's computing environment.

Appendix A

Listing 1: Parameter Values

- Host name = DEVHP0
- Port ID = 1599
- Protocol = TCP
- TurboIMAGE database name = tidb
- TurboIMAGE home group and account = gtwdb.tiacct
- User logon name(s) for tiacct = mgr.tiacct
- Mgr.Tiacct user/acct passwords = jack/jill
- SQL Database Environment (DBE) name = tidbe
- SID_NAME (in the “listener.ora” file) = tidb
- Gateway_sid (in the “listener.ora” file) = bucket
- SID (in the “tnsnames.ora”) = tidb
- Connect-String/Alias (in the “tnsnames.ora” file) = bucketalias
- Oracle Software Acct = oracle7
- User logon name(s) for ORACLE7 = mgr.oracle7
- Oracle Database Acct = oracledb
- User logon name(s) for ORACLEDB = mgr.oracledb
- Database link name = gtwlink
- Oracle database home = db.oracledb
- Oracle SID = F45

Listing 2: Sample Configuration Files

***initbucket.ora* file**

```
list db_name = bucket
sqldbe = 'tidbe.gtwdb.tiacct'
blocksize=50
sql_trace=false
resolve_binds=true
```

***listener.ora* file**

```
LISTENER = (ADDRESS_LIST =
             (ADDRESS =
              (PROTOCOL = TCP)
              (PORT = 1599)
              (HOST = DEVHPO)
             )
            )
STD_LIST_LISTENER = (SID_LIST =
                     (SID_DESC =
                      (ORACLE_HOME = ORACLE7)
                      (PROGRAM = GTWEX)
                      (ENV = (GATEWAY_SID = bucket)
                      )
                     )
                     (SID_DESC =
                      (SID_NAME = F45)
                      (ORACLE_HOME = DB.ORACLEDB)
                      )
                     )
                    )
```

***tnsnames.ora* file**

```
bucketalias =
  (DESCRIPTION =
   (ADDRESS =
    (PROTOCOL = TCP)
    (HOST = DEVHPO)
    (PORT = 1599)
   )
   (CONNECT_DATA = (SID = tidb))
  )
oraalias =
  (DESCRIPTION =
   (ADDRESS =
    (PROTOCOL = TCP)
    (HOST = DEVHPO)
    (PORT = 1599)
   )
   (CONNECT_DATA = (SID = F45) )
  )
```


Technical information in this document is
subject to change without notice.

Copyright Hewlett-Packard Company 1997
All Rights Reserved. Reproduction, adapta-
tion, or translation without prior written
permission is prohibited except as allowed
under the copyright laws.

Printed in USA RO0897
5966-0855E