HPWORLD 1996 Paper# 3009

Database Gateway Use In Heterogeneous Environments

by
Jennie Hou

Hewlett-Packard Company
19447 Pruneridge Avenue, MS 47UY
Cupertino, CA 95014

The traditional HP3000 based application often utilizes TurboIMAGE database management system as a data source. As applications grew or new applications were added, either modifications were made to existing TurboIMAGE database or new databases were 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, often the data source for these applications is a relational database management system residing either on the HP3000 or another platform. These new applications frequently require access to existing HP3000 data sources, the TurboIMAGE databases. Until recently, that access was accomplished through user application programming. Now, that access can be provided by database gateways from various vendors, such as HP, Oracle, Information Builders, and Sybase.

This paper examines the characteristics of database gateways to HP3000 data sources. Using the Oracle Transparent Gateway for IMAGE/SQL as a primary model, gateway architecture, gateway setup/usage, and performance tuning tips are discussed.

Database Gateway Use In Heterogeneous Environments
3009-1

WHAT IS A GATEWAY

Gateways are powerful coexistence tools which facilitate enterprise-wide information management by providing access to heterogeneous data sources residing on heterogeneous platforms at distributed locations. Gateways integrate various data sources and create the appearence 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. It provides operating system transparency where users do not need to know the operating system on which the data resides. It provides geographic transparency where 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. It provides network transparency allowing users to access the data regardless of the network protocol(s). It provides data storage transparency permitting 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, a user, in a single request, can integrate order information from Oracle7 on a HP9000 with manufacturing parts information from a TurboIMAGE database on HP3000 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 having to do extensive programming and customization in-house, each gateway product enables complete and ready-to-use solutions where 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 follow:

- Provide read/write data access and enable ad-hoc queries to heterogeneous databases using SQL.

- Provide easy data migration where data moving from one source to another data source can be accomplished with SQL statements.

- Allow users to leverage applications from their existing environment by allowing connectivity between HP3000 databases and other databases on other platforms. The ability to interoperate and exchange data is critical to the computing success of a business.

- Provide security based on combination of specified database servers and native operating system mechanisms.

- Use of client/server architecture where the user can select the most cost-effective platform to serve as the client, using in-house or vendors' tools and applications to access the desired platform as the server.

GATEWAY VENDORS FOR HP3000

There are multiple vendors that provide gateway products which can interface with the HP3000 platform and other UNIX-based platforms. Some of these vendors include Hewlett-Packard, Oracle, Information Builders, and Sybase. Below are brief descriptions of their products.

HP Allbase/Net

ALLBASE/NET from Hewlett-Packard is a data access product that provides transparent, remote access to TurboIMAGE and ALLBASE/SQL data as though it were stored locally. It uses a client/server architecture. The application runs on either a HP3000 or HP9000 client; the server can be either an HP3000 running TurboIMAGE or ALLBASE/SQL or an HP9000 running ALLBASE/SQL. ALLBASE/NET utilizes TCP/IP for network access.

ALLBASE/NET is bundled with ALLBASE/SQL and IMAGE/SQL on both HP3000 and HP9000 platforms. It allows queries or updates to TurboIMAGE and ALLBASE/SQL databases anywhere on the network using SQL calls. This solution provides economic 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

Oracle Transparent Gateway for IMAGE/SQL allows Oracle database users from more than 85 platforms 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 HP3000. This gateway provides data access via SQL to TurboIMAGE data which has been defined, mapped, and attached to IMAGE/SQL's database environment (DBE). This gateway makes TurboIMAGE's underlying network data model appears relational to 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 an TurboIMAGE data source. The gateway supports the TCP/IP protocol, and Oracle SQL*NET provides connectivity to the HP3000. 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 to heterogeneous database management systems on heterogeneous platforms transparently. On the HP3000, it allows TurboIMAGE, ALLBASE/SQL, ORACLE, and KSAM data to be retrieved from a

Database Gateway Use In Heterogeneous Environments
3009-4

multi-platform and multi-database environment. EDA/SQL utilizes TCP/IP or NETIPC for network access. In addition, EDA/SQL allows HP3000 users to access data on many other platforms. IBI also provides a window-based desktop decision support and analysis tool called FOCUS and which works in conjunction with the EDA/SQL product.

Sybase Open Client/Open Server & Direct Connect Gateway

Sybase and Proactive Systems are soon to come out with the Open Client/Open Server and Direct Connect Gateway products for the HP3000 platform. These Sybase products provide a database interoperability solution for the HP3000. They enable HP3000 users to read and write data to TurboIMAGE and ALLBASE/SQL from a Sybase environment or visa versa.

Gateways Illustration

This illustration depicts various ways for accessing data between HP3000, HP9000, and other platforms. Database connectivity can be done with multiple gateway products. The ability to coexist among heterogeneous platforms allows MIS flexibility to utilize each platform's strength.

ORACLE TRANSPARENT GATEWAY ARCHTECTURE

As mentioned in the introduction, the Oracle Transparent Gateway for IMAGE/SQL will be used as a primary model when discussing architecture, setup/usage, and performance tuning.

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

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

The illustration below depicts the architecture layer:

Processes On The Server & Client Nodes

What needs to be done on the server node and the client node when setting up and using the gateway and SQL*NET V2? What processes take place? Below are the major steps that occur in such scenario:

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.

   The child process "TNSLSNR" then adopts itself to "PROGEN", and it then becomes a system process.

   Now, the "TNSLSNR" process is waiting for connection requests to come. It's listening on the port specified in the "listener.ora" configuration file.

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.

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, protocol are provided as well as alias to be used as connect strings in database links.

4. Client Node Fires Off A Request

Let's say that 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.

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
- parses the request and determines if the request is for the Gateway or for 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 message to the client with the new port information.

6. Client Node Request Processed

Let's say 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. This file resides in the /<ORACCT>/GTW/admin directory. This file provides the name of the IMAGE/SQL DBE and the GATEWAY_SID name. The GTWEX is mapped to the IMAGE/SQL DBE, which in turns points to the physical TurboIMAGE database.

Database Gateway Use In Heterogeneous Environments
3009-8

At this point, the gateway to TurboIMAGE database is set.

7. Set Up The Database Links On The Client Node

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

Example:

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

8. Access TurboIMAGE Data From The Client Node

At this point, the client knows how to access the TurboIMAGE data from the Oracle database. Let's try issuing a SQL statement:

sql> select * from <TuboImageDB>.<Table>@<linkname>

example: select * from TIDB.EMPLOYEES@GTWLINK

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

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.

Database Gateway Use In Heterogeneous Environments
3009-9

SETUP STEPS

From the previous section, the major process steps are described. Using these steps as the framework, let us set up a gateway between a TurboIMAGE database and a Oracle database on a HP3000 system.

1. Configuration Files Setup

Set up the SQL*NET V2 configuration files. On the server node, the required configuration files are "listener.ora" and "sqlnet.ora". On the client node, the required configuration file is "tnsname.ora". In this example, the server node and the client node both reside on the same hardware system; thus, all above files will reside in the same Oracle Software Account <ORAACCT>. The default directory containing these files is /<ORAACCT>/NETWORK/admin. However, you can place these files anywhere on the system as long you set the "TNS_ADMIN" environment variable accordingly.

Set up the gateway initialization file, "init<gateway_sid>.ora". This file ALWAYS resides in the /<ORAACCT>/GTW/admin directory.

Gotchas

The key things to watch out for 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 gateway initialization file, the db_name is same as the gateway_sid name in the "listener.ora" file. Also, the name of the initialization file is "init<gateway_sid>.ora".

  Example: If the gateway_sid is "bucket", then the initialization filename is "initbucket.ora".

- In the "listener.ora" file,
  - o the gateway_sid name must match the db_name in the gateway initialization file.

Database Gateway Use In Heterogeneous Environments
3009-10

  o one SID_NAME entry must be the name of
    your TurboIMAGE database name.
  o the ORACLE_HOME must be set to your
    Oracle Software Account.
  o the PROGRAM must be set to "GTWEX".

- In the tnsname.ora file,
  o the SID name must be the name of your TurboIMAGE
    database.
  o 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.

Parameter Values

Below is a list of parameters and their values. They
are used in the examples throughout this section.

- 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

Sample Configuration Files

Here are some sample *.ora files:

initbucket.ora file
----
list
db_name = bucket
sqldbe = 'tidbe.gtwdb.tiacct'
blocksize=50
sql_trace=false
resolve_binds=true

Database Gateway Use In Heterogeneous Environments
3009-11

listener.ora file

```
LISTENER = (ADDRESS_LIST =
            (ADDRESS =
             (PROTOCOL = TCP)
             (PORT = 1599)
             (HOST = DEVHP0)
            )
           )
STD_LIST_LISTENER = (SID_LIST =
                     (SID_DESC =
                          (SID_NAME = tidb)
                          (ORACLE_HOME = ORACLE7)
                          (PROGRAM = GTWEX)
                          (ENV = (GATEWAY_SID = bucket) )
                     )
                    )
```

tnsnames.ora file

```
bucketalias =
 (DESCRIPTION =
  (ADDRESS =
   (PROTOCOL = TCP)
   (HOST = DEVHP0)
   (PORT = 1599)
  )
  (CONNECT_DATA = (SID = tidb))
 )
```

2. TNS_ADMIN Environment Variable Setup

Set up the "TNS_ADMIN" environment variable. This points to the location of the *.ora files 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 listner.

CI: lsnrctl.pub.<ORAACCT> start
Example: lsnrctl.pub.oracle7 start

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

```
CI:  gtwctl.pub.<ORAACCT>  r
Example:  gtwctl.pub.oracle7 r
```

5. IMAGE/SQL Mapping

Map your TurboIMAGE database to IMAGE/SQL on the server node.

```
CI:  imagesql.pub.sys
>>   set turbodb <TurboIMAGE database name>
        (Example:  set turbodb tidb)
>>   set sqldbe <dbe name>
        (Example:  set sqldbe 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. Mapping Verification

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. Oracle Database Link Creation

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 your 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 you choose 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. TurboIMAGE Data Access Through Oracle

At this point, you're 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:  select * from  tidb.employees@gtwlink
```

Database Gateway Use In Heterogeneous Environments
3009-14

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 TurboIMAGE rollback limitation and how best to work with it. The third case addresses 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, can and 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 currently designed to use ONLY 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. In IMAGE/SQL though, the users can override the default "RR" mode by specifying other modes at the beginning of a transaction. This option to choose other transaction modes may become available to the Oracle Transparent Gateway for IMAGE/SQL in later releases.

So what can be done to avoid unnecessary 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 you set are compatible with your existing applications.) With READ ONLY, no locking will occur.

If both READ ONLY capability and READ/WRITE capability are needed, one 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 an unique database link through Oracle SQLDBA. By doing this, the gateway will not obtain locks for the READ ONLY user. That user can have unlimited access to the TurboIMAGE data via gateway as long 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

Database Gateway Use In Heterogeneous Environments
3009-15

will occur even during reading. Thus, the locking hits would be much higher.

Another tip is that if the DBOPEN mode is set to READ/WRITE, 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 rollback a transaction if needed. However, the problem with the rollback capability is the limit placed on the rollback log buffer size. The transaction manager limits the buffer size to 4MB. What this means is that if a user wants to commit a transaction which is greater than 4MB, the transaction can't be completed. Unlike in TurboIMAGE, where 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.

Therefore, via the gateway to IMAGE/SQL to TurboIMAGE, this limitation is passed on to Oracle. To work around this limitation, it is best to divide the workload to be smaller than 4MB. This way, the limitation will not occur.

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 much quickly. The tip is to try to stay connected as long as possible versus connecting and disconnecting often.

CASE 4: Third-Party Indexing Tools

There are several third-party indexing tools for the TurboIMAGE database. The Oracle Transparent Gateway for IMAGE/SQL will be able to support these tools when IMAGE/SQL product can support them. Use of these third party indexing tools can improve the search performance.

Database Gateway Use In Heterogeneous Environments
3009-16

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. As mentioned in this paper, there are several vendors that can provide this technology to the HP3000 users. Although these gateways may vary architecturally, they do share some common attributes such as providing many levels of transparency, providing turnkey solution for end-users, using of client/server architecture, and allowing users to leverage their existing applications.

Setting up a gateway, as in the Oracle Transparent Gateway for IMAGE/SQl example, involves a series of steps on both the server node as well as the client node. Once the gateway is configured, TurboIMAGE data on HP3000 can be integrated into an industry-standard relational database, such as Oracle, and can be accessed via Structured Query Language (SQL) commands.

The overall gateway performance can be tuned. The user needs to be aware of the limitations set by the gateway used as well as IMAGE/SQL's limitations.

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

Database Gateway Use In Heterogeneous Environments
3009-17