Oracle Database migration between HP 9000 and HP Integrity platforms under HP-UX 11i

Executive summary

Migration of an Oracle® database in either direction between a PA-RISC-based HP 9000 server and an Intel® Itanium®-based HP Integrity server is a very simple process, thanks to the compatibility between HP-UX for the HP 9000 platform and HP-UX for the HP Integrity platform. This paper, which is targeted at IT professionals with HP-UX and Oracle administration expertise, documents the migration process and includes several sample migration scenarios with step-by-step instructions. Also included are results of performance testing that show no discernible performance difference between a database migrated from the HP 9000 platform and a database created "natively" on an HP Integrity server.

Target audience: Though we do discuss some of the steps involved in performing such activities as installing Oracle and building HP-UX Logical Volume Manager (LVM) volume groups, this paper is not intended to be a tutorial on those subjects. We assume that anyone using this paper will have sufficient expertise to understand the Oracle and HP-UX concepts that we discuss:

- **HP-UX administration:** administering the volume manager(s) used in your environment (for example, creating new disk or volume groups and new volumes), account/user administration, HP-UX software/patch installation, and remote-login administration (/etc/rhosts and .rhosts)
- **Oracle database:** installing, configuring, and backing up and restoring a database

Introduction

More than a decade ago, HP began working on a new computer architecture to serve as a follow-up to its highly successful PA-RISC architecture. HP called this architecture EPIC (Explicitly Parallel Instruction Computing) and, in 1994, partnered with Intel to create Intel's new 64-bit Itanium architecture. Today, HP offers its HP 9000 product family, based on the PA-RISC architecture, as well as its HP Integrity product family, based on the Intel Itanium processor. These two product families support a common UNIX® operating system, HP-UX 11i version 2 (11i v2) or the recently released version 3 (11i v3).

HP's Oracle lab has extensively tested the process of migrating an Oracle database in either direction between a PA-RISC-based HP-UX environment (HP 9000) and an HP Integrity server environment running HP-UX. This paper, the result of that testing, is intended to be used as a guide for anyone who wishes to migrate an Oracle database between the HP 9000 and HP Integrity platforms (in either direction).

In this paper we will discuss the direct result of testing that we've done in the lab for Oracle database migration scenarios on HP-UX (including Oracle Real Application Clusters, i.e., RAC). We will not discuss the migration of applications other than the Oracle database, nor will we discuss migration in non-HP-UX environments (such as Tru64 UNIX or Linux).

Migration made even easier

One of the major goals HP had in co-designing the Intel Itanium architecture was to maintain a high degree of compatibility with the PA-RISC-based HP-UX environment. One result of this compatibility is that a database created on an HP 9000 system will not need any sort of data conversion in order to run properly on an Integrity system. Migrating an Oracle database in an HP-UX environment from PA-RISC to the Itanium architecture will thus be a simple process. With the right level of preparation, as shown below, this can be as simple as shutting down the database on the source HP-UX server and bringing it back up on the target HP-UX server.

The original version of this white paper, Oracle9i Release 2 Database Migration from HP-UX 11.0 or HP-UX 11i on HP 9000 to HP-UX 11i v2 on HP Integrity, was written to document upward migration

from the HP 9000 family to the HP Integrity family. Since then, we've received feedback from customers intending to maintain both Integrity and HP 9000 servers in their data centers that they would benefit from the option of migrating in the opposite direction, from Integrity to PA-RISC. We're thus pleased to announce that migration between the PA-RISC and Itanium architectures in either direction is now supported (and documented here).

If the thought of a mixed IT shop with both HP 9000 and Integrity servers makes you curious about the possibility of mixed-architecture high-availability environments, we encourage you to see the white paper entitled "Failover of Oracle Database between HP 9000 (PA-RISC) servers and HP Integrity (IPF) servers under HP-UX 11i", which discusses all manner of failover scenarios involving mixed environments.

Minimize downtime

As you'll see in the following sections, there is no need to do any data conversion to migrate the database. This is of great advantage not only for ease of migration but also for minimizing downtime. The exact same database can be used on either HP 9000 servers or HP Integrity servers. A simple database shutdown on the source server and then a startup on the target server – or a backup on one and restore on the other – is all that is needed. You know your environment best and which process makes the most sense. HP assures the data compatibility, and that leaves you with the flexibility to use whichever method suits your environment best.

HP-UX 11i release names and release identifiers

Each HP-UX 11i release has an associated version number and release identifier; in this document, HP-UX versions are often referenced by their shorthand names (release identifiers). The following table shows the releases available for HP-UX 11i.

To determine the release of the operating system running on a particular server, use the uname (1) command with the -r option.

Table 1. HP-UX 11 releases

Version number	Release identifier (shorthand name)	Supported processor architecture
B.11.00	HP-UX 11.0	HP 9000 only
B.11.11	HP-UX 11i v1	HP 9000 only
B.11.20	HP-UX 11i v1.5	HP Integrity only
B.11.22	HP-UX 11i v1.6	HP Integrity only
B.11.23 *	HP-UX 11i v2 (original Sept 2003 release)	HP Integrity only
B.11.23 *	HP-UX 11i v2 September 2004 (or later)	HP 9000 or Integrity
B.11.31	HP-UX 11i v3	HP 9000 or Integrity

* HP-UX 11i v2 was originally released in September of 2003 for HP Integrity systems. A newer version of HP-UX 11i v2, released a year later and known as the September 2004 update, was available for either HP Integrity or HP 9000 systems. Both versions of HP-UX 11i v2 use the B.11.23 version number. The original version is known simply as HP-UX 11i v2, while the updated version is called HP-UX 11i v2 September 2004.

To check and see if an Integrity system has HP-UX 11i v2 September 2004 or later loaded on it; use the following swlist command to check the version of BUNDLE11i:

```
swlist –l bundle | grep BUNDLE11i
```

The version of BUNDLE11i for the September 2004 update is B.11.23.0409.3. If version B.11.23.0408.1 is on your system, you do not have the September 2004 update of HP-UX 11i v2. (There is no need to check your 11i v2 HP 9000 system since only the September 2004 version of 11i v2 is supported on the HP 9000 platform.)

For more information about HP-UX 11i v2 September 2004, go to: http://www1.itrc.hp.com/service/rsb/rsbDisplay.do?fileName=patches_hpux/hpux11iv2_9000_integrity.htm

The "March 2007" patch bundle for HP-UX 11i v2, which is specified below as the minimum OS requirement for Oracle Database 11g, can be identified by using SWLIST as described above – the bundle identifier is B.11.23.0703.

Migration prerequisites

The following paragraphs explain the migration prerequisites – these are summarized in the table which follows. The descriptions and the table refer to two migration methods which we call "hot" and "cold" – these are fully defined in the Migration summary section, below. **In all cases, the specified HP-UX versions are the minimum required – any later versions or releases which are supported for the specified Oracle release are also supported for migration purposes.** For example: the notation "11i v1 (11.11) or 11i v2 (11.23) or greater" would mean that 11i v1 (11.11) is supported, 11.22 is not supported, and 11i v2 (11.23) and any later OS release (including, for example, 11i v3 [11.31]) are supported. And "at least 11i v2 September 2004" would indicate that subsequent releases of 11i v2 are also supported, as well as subsequent HP-UX releases such as 11i v3.

To migrate an Oracle database ***from*** **an HP 9000 server** ***to*** **an HP Integrity server** running HP-UX, one of the following combinations of Oracle and HP-UX versions is required:

- Oracle Database 11g (all versions). Any HP-UX version which supports Oracle 11g (i.e., 11i v2 with at least the March 2007 patch bundle, or 11i v3) is supported for migration purposes. Be sure to consult the installation guide for Oracle 11g and install any specified OS patches on both platforms.
- Oracle Database 10g Release 2 ("10gR2", version 10.2.0.2 or higher). The operating system on the HP 9000 must be HP-UX 11.11, 11i v2 (September 2004), or greater. On the Integrity server, HP-UX 11i v2 September 2004 or greater is required. Consult the installation guide for Oracle 10gR2 and install any specified OS patches on both platforms.
- Oracle9i Database Release 2 ("9iR2", version 9.2.0.7 or later). HP-UX version must be 11i v1 (11.11) or 11i v2 (11.23) or later on the HP 9000 platform and at least "11i v2 September 2004" (11.23) on the HP Integrity platform.
- Oracle9i Database Release 2 ("9iR2", version 9.2.0.2 or later). HP-UX version must be 11.0 or later on the HP 9000 platform and at least 11i v2 (11.23) on the HP Integrity platform. Only the "cold" migration method (moving or copying files) has been tested and is supported for 9.2.0.2 migration.
- Oracle8i Database (8.1.7): an Oracle8i (version 8.1.7) database can be migrated from an HP 9000 server to an HP Integrity system, but it **must** be upgraded to Oracle9i release 2 before it can be opened on the HP Integrity system. Oracle8i is not supported on the HP Integrity platform. The HP-UX version must be at least 11.0 on the HP 9000 platform and at least 11i v2 (11.23) on the HP Integrity platform. Only the "cold" migration method (moving or copying files) has been tested and is supported for 8*i* migration.

  (Note: if your 8.1.7 database is running on a 32-bit platform, see http://download.oracle.com/docs/cd/B10501_01/server.920/a96530/migintro.htm#1006950)

To migrate an Oracle database ***from*** **an HP Integrity server** running HP-UX ***to*** **an HP 9000 server**, one of the following combinations of Oracle and HP-UX versions is required:

- Oracle Database 11g (any release/version) or 10g Release 2 ("10gR2", version 10.2.0.2 or higher). The operating system on both the source and target systems must be at least the September 2004 update of HP-UX 11i Version 2 ("11i v2 September 2004.") – see the HP-UX release chart (above) for instructions how to verify installation of the September 2004 update of 11i v2. Be sure to consult the installation guide for Oracle 10gR2 and install any specified OS patches on both platforms.
- Oracle9i Database Release 2 ("9iR2", version 9.2.0.7 or higher). HP-UX version must be at least 11i v2 (11.23) – with the September 2004 update patches (and any other OS patches specified in the release notes and installation guide for 10gR2) – installed on both platforms. (Note: yes, the OS patches specified in the Oracle 10gR2 installation documentation are required for migration of a 9iR2 database.)

Oracle Database versions not listed here (any version prior to 8.1.7, 9i versions prior to 9.2.0.2, or 10g versions prior to 10.2.0.2) are not supported for cross-architectural migration.

Note: Now that migration is supported in either direction across the two architectural platforms, keep in mind the general HP philosophy that OS compatibility is only provided when moving to an equivalent or a newer OS version. Thus, for example, while we might support database migration from an HP 9000 running HP-UX 11i v1 to an HP Integrity server running HP-UX 11i v2, a downward migration (11i v2 to 11i v1) is not supported regardless of the platforms (this is true even between two systems on the same platform). Migration from an 11i v2-based system can only be to a system running 11i v2 or greater. **Thus, do not rely on reverse migration as a fallback if you would need to migrate your database back to a lesser or earlier HP-UX version.**

Table 2. Migration prerequisites summarized (minimum required HP-UX/Oracle versions)

ORACLE VERSION	HP 9000 TO INTEGRITY	INTEGRITY TO HP 9000
8i	COLD MIGRATION Oracle 8i - 8.1.7 HP 9000 – 11.0 HP Integrity – 11i v2 (11.23) (Upgrade 8i database to 9*i* before opening)	NOT POSSIBLE (8i not supported on Integrity)
9*i*	COLD MIGRATION Oracle 9iR2 - 9.2.0.2 or higher HP 9000 – 11.0 HP Integrity – 11i v2 (11.23) HOT MIGRATION Oracle 9iR2 - 9.2.0.7 or higher HP 9000: 11i v1 (11.11) or 11i v2 (11.23*) † HP Integrity – 11i v2 (11.23*)	COLD OR HOT MIGRATION Oracle 9iR2 - 9.2.0.7 or higher HP Integrity – 11i v2 (11.23*) † HP 9000 – 11i v2 (11.23*) †
10*g*	COLD OR HOT MIGRATION Oracle 10gR2 - 10.2.0.2 or higher HP 9000 – 11i v1 (11.11) or 11i v2 (11.23*) † HP Integrity – 11i v2 (11.23*) †	COLD OR HOT MIGRATION Oracle 10gR2 - 10.2.0.2 or higher HP Integrity – 11i v2 (11.23*)† HP 9000 – 11i v2 (11.23*) †
11*g*	COLD OR HOT MIGRATION Oracle 11g – any version HP 9000 – 11i v2 (11.23*) † HP Integrity – 11i v2 (11.23*) †	COLD OR HOT MIGRATION Oracle 11g – any version HP Integrity – 11i v2 (11.23*)† HP 9000 – 11i v2 (11.23*) †

HP 9000 and HP Integrity numbers refer to the version of the HP-UX operating system (in each case, any subsequent OS versions which support the specified Oracle version are also supported). Reminder: in every case, in addition to the requirements in the above table, the target server must be running at least the same HP-UX version as the source server.

* 11i v2 (11.23) September 2004 (or later) release. The earlier 11i v2 release, only available for HP Integrity, does not support any of these migration options EXCEPT as the target of an Oracle 9i or 8i cold migration.

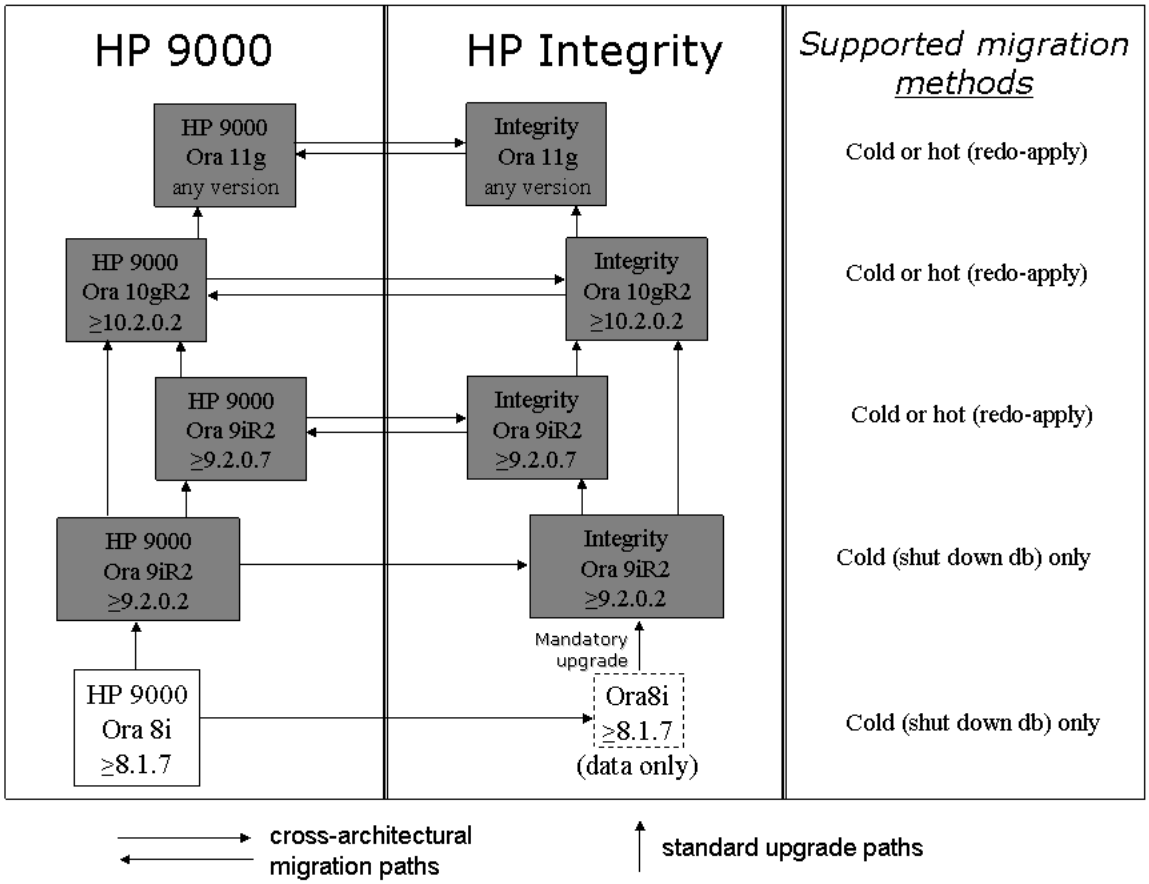
See HP-UX 11i release names and release identifiers, above, for a formal definition of "11i v2 September 2004". (Note: For Oracle Database 11g running under HP-UX 11i v2, the March 2007 patch bundle is required – see the Oracle Database 11g installation documentation.)

† Patch HP 9000 and Integrity OS following recommendations of 10gR2 Installation Guide for a 10g or 9i migration (yes, even for a 9i migration), or the 11g Installation Guide for an 11g migration.

Note that database migration is agnostic with respect to the format the database is stored in. Thus, the database could be based on Oracle ASM (Automatic Storage Management) or it could be in file systems or raw devices, with or without HP Logical Volume Manager, etc.; the migration is supported as long as the database's format is supported on both source and target environments, and all other requirements and restrictions are met.

Migration summary

Figure 1. Supported migration paths and methods (see Table 2 for OS requirements)



Migration of an Oracle database between the HP 9000 and HP Integrity families is almost as simple as migrating between two systems in the same family – as long as the prerequisites (see above) are met, the migration process differs only to the extent that we need to make an accommodation for differences between the native object code of the two architectures. Thus, while we'll need to make sure to install the appropriate PA-RISC or Itanium versions of the Oracle software (and we'll need to recompile any natively compiled database objects – discussed later), we will not have to perform any data conversions of any kind.

(Oracle 8i database migration is supported from the HP 9000 to the HP Integrity platform only, but since Oracle 8i is not supported on HP Integrity, the database must be upgraded to at least Oracle 9*i* following the migration. Thus, someone wishing to migrate an Oracle 8i database must upgrade it to at least Oracle 9*i*, but they may choose whether to perform the upgrade before the migration, on the HP 9000 system, or after the migration, on the HP Integrity system.)

The migration process can thus be summarized as follows:

- Ensure that the Oracle database version and HP-UX version on the source system are one of the supported combinations listed in the prerequisite section above; upgrade if necessary, applying all appropriate patches.

- For the target system, select the appropriate combination of Oracle and HP-UX versions per the prerequisite list above. Prepare the target system, including HP-UX and Oracle installation (be sure to install the version of Oracle that is specific to the platform – either Integrity or HP 9000), according to the proper installation procedures; this includes using the appropriate volume manager procedures to create the disk/volume structure (and file systems, if necessary) to store your database.
- Using the method of your choice, make the data available to the target system. (Always make a backup copy of your database before doing any major changes!)
  - **Hot migration**: The database can be backed up on the source system using standard, supported methods for backing up an Oracle database, and restored on the target system. The backup-restore method has the advantage of permitting an "almost-zero-downtime migration" since the database could be backed up in "online" mode and continue to operate on the source system until the target environment is ready to be put in production, with the redo logs which accumulated on the source system applied to the database on the target system. In this case, disk array-based copy methodologies such as HP StorageWorks Business Copy or Continuous Access could be used to propagate the active redo logs to the target system and mirror the live log updates from the source system to the target. An "offline" backup/restore is of course also possible. (Note: be sure to follow proper Oracle backup procedures or you may wind up with a backup that is not restorable. Remember to use Oracle's RMAN or the "begin backup" and "end backup" constructs if you are doing an online backup.)
  - **Cold migration**: The database instance can be shut down, the database files (including control files, undo datafiles, online redo logs, Oracle initialization file, etc.) copied or moved to the target system, and then the database instance started up again in the new environment. Moving the database files is generally quicker and easier than copying, but copying allows you to keep the database in place on the source system as a backup. Business Copy, Continuous Access, or similar array-based tools could be employed to perform the actual copying of the database files – or they could be copied in a more traditional manner (for example, ftp or rcp). Moving the database files would be accomplished by moving the database disks themselves, either logically (by reconfiguring the SAN or changing the host-presentation of the database LUNs) or physically (for example, recabling – disconnect and reconnect from one system to another).
- Ensure that the database configuration files (for example, init.ora, listener.ora) reflect the target system's name and IP address. If necessary, regenerate the control files.
- Scan the database to locate any objects that were natively compiled for the source architecture, and recompile them to run on the target environment (see below for more information).
- If you migrated an Oracle 8i database from an HP 9000, upgrade the database to Oracle9i release 2.

These steps may differ slightly depending on the nature of your Oracle installation—the differences will be highlighted in the examples below. We've included instructions which will accommodate the following Oracle scenarios:

- **Single-instance Oracle** (non-clustered) implemented using **file systems** built on top of LVM logical volumes
- **Single-instance Oracle** (non-clustered) implemented using **raw volumes** (LVM logical volumes without file systems)
- **Clustered Oracle** (Oracle9i or 10g Real Application Cluster) implemented using **raw volumes** (LVM logical volumes without file systems)
- We will not consider the case of raw volumes implemented with so-called "hard partitions" (non-LVM raw disks) as this approach is very rarely used.

Hot Migration: Minimize downtime, assure data integrity for a live production database

We've listed a variety of migration methods and options in order to let you decide what's best in your own environment, but the "hot migration" method (backup, restore, and apply logs) stands out as being particularly suitable for use in a live production environment. This method minimizes downtime by permitting the source database to remain available right up until the database is ready for production on the target server, and has the advantage of a 'built-in' integrity check in the application of the accumulated source-system log files to the target-system database. Here's a simple summary of hot migration:

- Make sure archive logging is enabled, as it should be for any database whose data is important enough to warrant backing up.
- Using Oracle RMAN, perform an "online" backup of the source database. The database stays up and running and available to users during the backup and afterwards.
- After setting up the target system and installing Oracle, move or copy the backup files from the source system to the target system and use Oracle RMAN to restore the database onto the target system.
- Allow sufficient time to elapse to ensure that the source database, still in production, will experience a representative array of production transactions – the details of which will be logged in the source system's redo logs. Choose all or some (the oldest) of the logs created since your online backup was created and copy them to the target system.
- Apply the copied log files to the target database and monitor the alert logs for errors. Upon successful application of the copied logs, pull additional newer logs from the source system and apply them to the target database. The more logs you apply, the more current the target database will become, and – assuming the application succeeds without errors – the more confident you will be that the migration was successful.
- In preparation for going live on the target database, continue to copy and apply the logs as they accumulate on the source system. At switchover time, shut down the source database and copy and apply the last set of logfiles to the target system database. The target database will now be fully up-to-date and identical to the source database – and ready to be put into production.

Keeping the "old" environment available as a standby

Like many customers, you may see the advantage of maintaining your "old" server(s) as a standby environment which can be activated in case of a disaster or other problem with the new environment. With a little bit of extra planning (and the appropriate technology, such as HP StorageWorks Continuous Access, or Oracle Data Guard), you should be able to set up and test the infrastructure in advance, perform the database migration, and then keep the "old" database up to date as production changes are made to the "new" database.

If you are interested in setting up such a "mixed failover environment", be aware that there are some restrictions (for example, since data migration is never supported from a newer OS release to an older one, both the "old" and "new" environments must be running the same OS and sub-system version/patch-level) – for more details, be sure to read the white paper entitled "Failover of Oracle Database between HP 9000 (PA-RISC) servers and HP Integrity (IPF) servers under HP-UX 11i", which discusses all manner of failover scenarios involving mixed environments such as these.

"Migrated database" versus "native database" performance

Before getting into the details of the migration, you may be wondering how your existing database will perform on the new server. It has been well documented that application programs will perform better when they are recompiled into native object code, but does this "native advantage" apply to data as well? Is there a performance advantage in rebuilding an HP 9000 database so it is "native" on an HP Integrity system? Great news: the answer is no. There is no need to rebuild your database.

No data conversion takes place when moving an Oracle database between HP 9000 and HP Integrity platforms. The database structures and the database block layout are identical, so when you move your HP 9000 database to HP Integrity (or vice-versa), the database engine has no knowledge of the fact that it is operating on data that was created on another platform. There would be no noticeable performance benefit from going through the process of re-creating your database natively on the target system.

Laboratory performance comparisons

To compare the performance of a "native" and "migrated" database, some decision support system (DSS) workloads from a major standard DSS benchmark suite were used, involving several tests on the same system. These DSS workloads were chosen because they included more than 20 queries that stress the system in different ways (that is, I/O intensive, CPU intensive, etc.).

The following methodology was used:

1. The whole package was run natively on the HP Integrity system. That included creating the database, populating it, and running the queries.
2. The same database was created and populated on an HP 9000 environment and then migrated to the HP Integrity system, where the same query workload was run and measured.
3. Performance results achieved during steps 1 and 2 were compared.

In both cases, the queries were run and measured on the HP Integrity system. The only difference is that the database was created in different Oracle 9.2 environments (HP 9000 versus HP Integrity).

The following charts show the performance of two of the queries over time. In the interest of space, we didn't include all 22 queries since they all showed similar behavior. "Native" refers to the database created on the HP Integrity server; "Migrated" refers to the database created on an HP 9000 server and migrated to the HP Integrity server.

Figure 2. Run times of one type of transaction, native database versus migrated

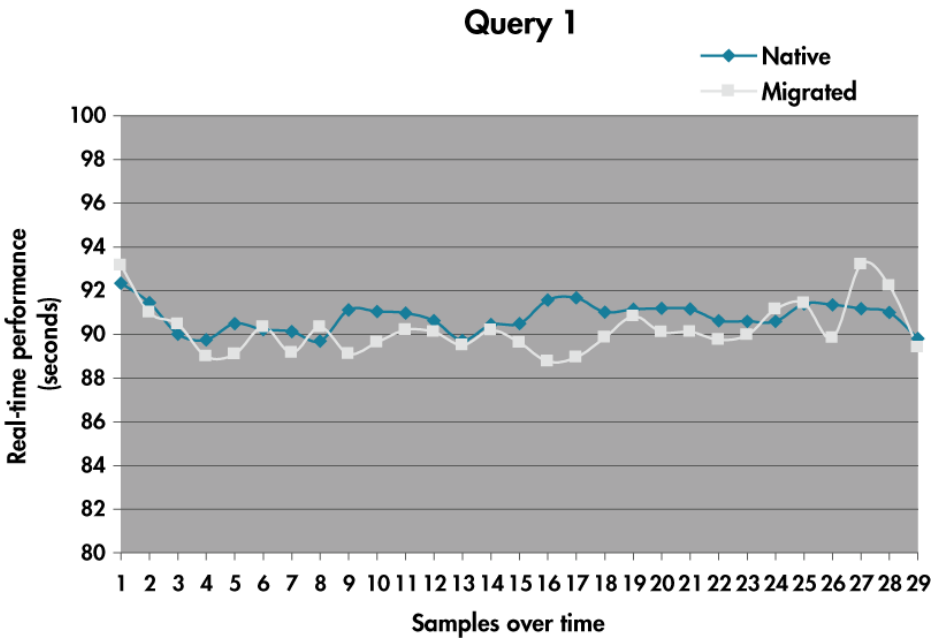
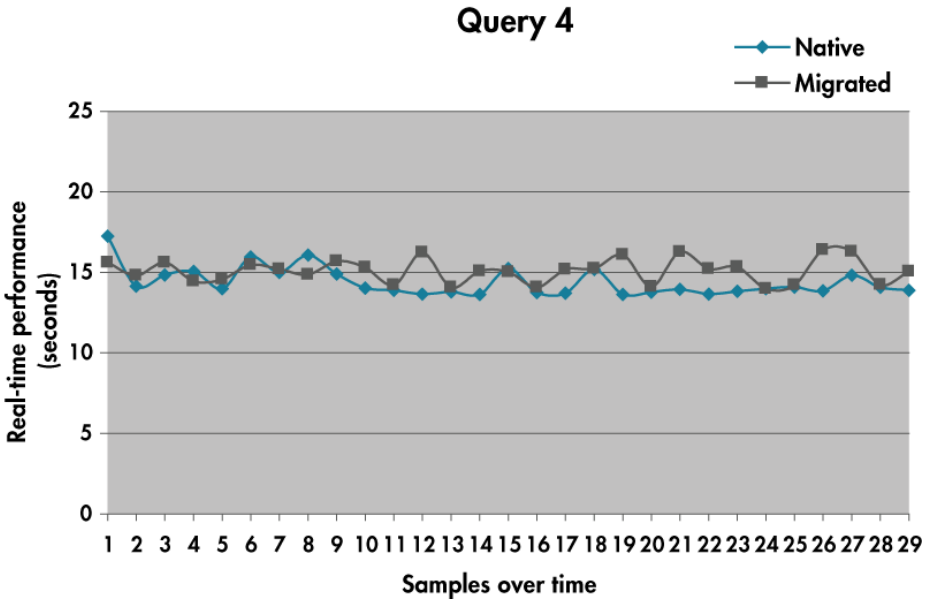


Figure 3. Run times of another type of transaction, native database versus migrated



While there was some fluctuation in the individual data points due to the random nature of these tests, the graphs show that overall there is very little, if any, performance difference between the "native" and "migrated" databases.

Incidentally, when HP ran the Oracle Applications Standard Benchmark, the results of which we published in early 2003, we used a migrated database on a four-processor HP Integrity rx5670 server. In the highly competitive world of industry-standard benchmarks, every effort is made to eliminate potential performance issues so we get the best possible result—and we certainly wouldn't have used a migrated database if there was a performance penalty associated with it.

Migration, step-by-step

The following step-by-step process suggests how an Oracle database could be migrated across architectures. These particular steps assume the use of HP-UX Logical Volume Manager (or, for a clustered solution, Shared Logical Volume Manager) but an equivalent process would be applicable for other supported volume managers.

Note: We've assumed an appropriate level of familiarity with HP-UX commands in the list that follows. If you need more detail than what's included here, check the man page for the command in question. **Unless otherwise specified, all the following HP-UX commands should be performed when logged in to the system as the root user.**

1. Consult the list of prerequisites (above) and ensure that the source and target servers meet the specified HP-UX version and patch requirements. Note the corresponding Oracle version requirements for your environment.
2. Ensure that the Oracle database software on the source system is compatible with the requirements specified in the prerequisites section (above). Upgrade to one of the supported Oracle releases if necessary.
3. Install the selected version of Oracle on the target environment, after preparing and configuring the target system(s) per the appropriate Oracle installation and configuration guide. This might include:
    1. Installing or upgrading to the appropriate version of HP-UX and applying recommended patches, if not already complete.
    2. Installing and configuring supporting products, such as HP Serviceguard Extension for Real Application Clusters (eRAC), that may be required or desired.
    3. Creating the volume (and, if appropriate, file) structure which will be used to hold the database. (This will not be necessary if you plan to migrate your database by moving the storage hardware to the target environment.)
4. If you've chosen to back up and restore your database to the target system, perform the backup now. (If you're going to shut down and copy or move the database files themselves, skip to the next step). To use an "online" backup, an Oracle-supported tool (e.g., RMAN) must be used; follow the standard published procedures for performing an online backup, then copy or move the resulting backup to the target system (see next step). While the migration steps are performed and the database is readied for production on the target system, the database on the source system can be kept in production mode; logfiles which are generated on the source system during this time should be archived for application to the migrated database prior to going live on the target system. Alternatively, you can use an offline backup/restore method: shut down the database and perform a "cold" backup, then move or copy the backup to the target system (see next step). Of course, the usual storage-based techniques – for example, volume-manager mirror-and-split, or the use of a disk array's "Business Copy" feature – can be used to facilitate the Oracle backup process.
5. Copy or move your database (or your backup) to the target system(s). Select step (a) for copying, or step (b) if you're moving.
    1. If you're copying your database, shut down the database on the source system, and then copy the data over to the target system using whatever method is most appropriate for your environment and that you're most comfortable with. For example, you might prefer

to copy the data to tape and then restore the tape on the target system. If both the source and target systems are on the same network, you could use *dd* through a network pipe to copy your Oracle volumes to the target system, as in this example for an LVM-based system:

```
cd /dev/vgORA
dd if=rlv01 | remsh targetsysname "dd of=/dev/vgORA/rlv01"
```

(Repeat the *dd* command for each additional logical volume to be copied.)

You could also use a network pipe to copy individual data files. For example, to copy all the files from the "/oradb" directory on the source system to the "/oradb" directory on the target system, issue the following commands while logged in to the source system (be sure to type these correctly—especially the blank spaces which delimit the parentheses):

```
cd /oradb
tar cf – . | ( remsh targetsysname cd /oradb  \; tar xf – )
```

Be sure to copy the critical incidental Oracle files such as the control files, init.ora files, listener.ora and tnsnames.ora, etc.

2. If you're moving an LVM-based database, follow these steps. (For other volume managers, substitute the equivalent volume-management procedures.)
    i. Shut down the database. (Now would be a really good time to make a backup copy of your database.)
    ii. For a file-system-based database, unmount the file system(s) using the *umount* command.
    iii. Deactivate the volume group(s) using the *vgchange* command.
    iv. Create a map file, which will facilitate moving the volume group(s) to the target system, by using the *vgexport* command. Use the *–p* (preview) option to avoid expunging the volume group from the source system at this time. We also recommend using the *–s* (shared) option, which was designed to simplify the process of moving volume groups between the nodes of a clustered system, but which is also available for non-clustered systems. Here's an example of a *vgexport* command which will create a map file named 'mapfile' for the volume group 'vgORA':

        ```
        vgexport –p –m mapfile –s /dev/vgORA
        ```

        Copy the map file to the target system (where it will be used by a *vgimport* command to re-create the volume group).
    v. Provide the target system access to the disks containing the database. This could require physically recabling the disks from the source system to the target system. It could also be accomplished by reconfiguring your SAN, or (in the case of a disk array) reconfiguring the LUNs to be visible to the target system. On the target server, perform an *ioscan* and then an *insf* to initiate configuration of the newly added disks or LUNs.
    vi. Import the volume group(s) using the map files and the volume group directories and group files you created earlier. For example, use:

        ```
        vgimport –m mapfile –s /dev/vgORA
        ```

    vii. Change the volume group special files to be owned by the same owner and group as on the source system, and ensure correct permissions; for example:

```
chown –R oracle:dba /dev/vgORA
chmod 644 /dev/vgORA/*
```

viii. Activate the volume group(s) using *vgchange –a y*. (For a clustered system using HP Shared Logical Volume Manager, you'll need to take the extra steps to change the volume groups to permit sharing, using *vgchange –c y –S y*, and then activate the volume group in shared mode using *vgchange –a s*.)

ix. If your database is implemented using file systems, mount the logical volumes contained in the volume groups you've just activated (e.g., *mount /dev/vgORA/lv01 /ora01*).

6. On the target system, edit the Oracle configuration files (for example, init.ora, listener.ora, etc.) to reflect the new IP address, hostname, pathnames, and other characteristics of the target system.
7. Log in to HP-UX as the oracle user and start up the database. (If this is a Real Application Cluster installation, verify that the RAC option is installed; *sqlplus* will show the phrase "With the Real Application Clusters option" in its banner as long as the database instance is running.)
8. If the database is Oracle8i version 8.1.7, then upgrade it to Oracle9i release 2 using Oracle's recommended upgrade procedures.
9. Recompile any natively compiled database objects. (See the section entitled Recompiling natively compiled objects at the end of this paper.

Sample Oracle Database 10gR2 migrations

For sample migrations of an Oracle 9i or 8i Database, see the earlier version of this paper, Oracle9i Release 2 Database Migration from HP-UX 11.0 or HP-UX 11i on HP 9000 to HP-UX 11i v2 on HP Integrity.

Single-instance Oracle on file systems or raw volumes

This is a recommended procedure for migrating either a file-system-based or raw-volume-based single-instance database between an HP 9000 server and an Integrity system in either direction – the instructions would apply equally in either case. We will thus refer to the two environments as "source" and "target".

The HP 9000 server is assumed to be a single-node server running the HP-UX 11i v1 operating system (11.11). Note: migration is only supported to an equivalent or better OS version, so if this HP 9000 is to be the target environment, it would have to be upgraded to at least the same HP-UX version running on the Integrity system. (See the note under the Migration prerequisites section above.)

The HP Integrity server is assumed to be a single-node server running HP-UX 11i v2 (11.23) September 2004.

The source environment, regardless of whether it is the HP 9000 or HP Integrity server, has the following characteristics:

- Oracle Database 10gR2 (version 10.2.0.2)
- Database built either on JFS (Journal File System) files on top of LVM logical volumes, or built using "raw volumes" (that is, built directly on LVM volumes without a file-system structure on top)

Prior to migration

If migrating from an HP Integrity server to an HP 9000 server, the OS on the HP 9000 must be upgraded from 11i v1 to at least 11i v2 September 2004. To migrate from the HP 9000 to Integrity, the HP 9000 may be allowed to continue running HP-UX 11.11 or it could be upgraded to 11i v2 (11.23) or greater

For a 10gR2 migration such as this one, the prerequisites include the requirement that all OS patches specified in the Oracle Database 10gR2 installation guide for HP-UX be applied. Thus, the Oracle Database 10gR2 installation manual for the source environment should be consulted, and all required OS patches applied.

Our Oracle Database version, 10.2.0.2, meets the requirements for a 10gR2 migration.

Target-system preparation

The target system should be set up in accordance with standard procedures for configuring and installing HP-UX for an Oracle database. The Oracle database installation documentation for HP-UX (for HP 9000 or HP Integrity servers) should be followed, including the specifications for OS kernel parameters and root-volume file system sizes.

Following OS installation and setup, the Oracle software (version 10.2.0.2) should be installed on the target system. In the Oracle-home directory on the target system, set up the shell's environment file (.profile) to create the ORACLE_SID environment variable using the same SID as on the source system. (In the following examples, we will use environment-variable notation such as "$oh-src" to refer to the Oracle home on the source server and "$oh-trgt" for the target server; you should substitute the actual paths when you type these commands.)

Using *rcp*, copy all the database configuration and parameter files (including the initSID.ora file) from the source system to the target system (assuming you are logged in to the target system):

```
rcp source-sys:$oh-src/dbs/*.ora $oh-trgt/dbs/.
rcp source-sys:$oh-src/dbs/orapwsid $oh-trgt/dbs/.
rcp -r source-sys:$oh-src/admin/sid* $oh-trgt/admin/.
```

The last command shown above copies the admin directory, which contains bdump, udump, etc.

Ensure that the access permissions and file owners for all these files and directories are the same on the target system as on the source system. (Improperly set file permissions would cause an ORA-01102 when opening the database.) Edit the files as appropriate for the target environment.

Copying or moving the database

Ordinarily, at this point one would have to choose between copying or moving the database to the target system. We will document the steps for each alternative below.

Alternative 1: Copying the database to the target system

By copying the database, you can leave the database intact on the source system, allowing for a fallback in case of problems. In fact, production can be resumed on the source system while the migration process is executed and the new database is prepared on the target system. Once the database on the target system is ready, the accumulated logfiles can be moved across from the source system and applied to the new database, which not only brings the new database current, but also provides a thorough and robust test of the new environment.

Before the database can be copied, the volume group(s) and logical volume(s) must be built on the target system (for a Symantec Veritas Volume Manager environment, the equivalent disk groups and volumes would have to be created). For a file-system-based database, once the volumes are created, the additional steps of creating the file system infrastructure are necessary (create mount points, build and mount file systems with the proper options, and create directory structures to hold the database files). Since these are all basic HP-UX tasks, we won't document them here.

Now, do you want to copy the database by copying the underlying files (or volumes) or by doing a full backup on the source system and a recovery on the target system? The backup/restore method permits the option of an online backup, which eliminates the need to shut down your production environment on the source system. The process of doing a backup/restore should be familiar to any Oracle database administrator and won't be documented here.

If you prefer to copy the database files or volumes to the target system, shut down the database instance on the source system (to ensure that the database files are not accessed while they're being copied). If your database is stored in a mirrored environment, you could split your mirrors at this point, and re-open the database and resume production on one side of the mirror while copying the database from the other side of the mirror to the target system. Otherwise, you'll need to keep your database instance shut down until the copy operation completes.

For a file-system-based database, use *rcp* to copy the files across the network. For example:

```
rcp –r source-sys:$oh-src/oradata/sid $oh-trgt/oradata/sid
```

For a database built on raw volumes, copy the individual logical volumes from the source to the target environment using dd across a network pipe. This example shows the operation being initiated from the source system:

```
cd /dev/vgORA
dd if=rlv01 | remsh target-sys "dd of=/dev/vgORA/rlv01"
```

Log out and log back in as the oracle user on the target system (to properly set the environment variables) and start up the database. You should now check for any natively compiled objects and recompile them for the target environment. (See the section entitled Recompiling natively compiled objects at the end of this paper.)

Alternative 2: Moving the database to the target system

By moving the database, we don't have to wait for the files to copy, but we don't have a database left on the source system in case of problems, and we have to suspend production from the time the database is shut down on the source system until it can be successfully restarted on the target system.

First, shut down the database on the source system. Next, log in to the source system as the root user. If the database is built on file systems, unmount the file system(s):

```
umount /opt/oracle/db
```

Now deactivate the volume group(s). For example:

```
vgchange –a n /dev/vgORA
```

Next, use the *vgexport* command to create a map file:

```
vgexport –p –s –m vg.map /dev/vgORA
```

Then, copy the map file to the target system.

Now it's time to move the disks to the target system. For a database stored on a SAN-based storage system, this could involve recabling the fibre channel cables to the target system, or adding the target system to the same SAN as the source system is connected to. Depending on the type of storage devices involved, you might have to reconfigure them to recognize and be accessible to the target environment rather than the source environment.

Once the storage is moved and/or reconfigured, proper HP-UX procedures for configuring new storage devices should be followed on the target environment (this may be as simple as an *ioscan* command, but disk array devices usually require array-discovery operations as well – see the documentation for your device).

After the storage devices are properly recognized by HP-UX, import the volume groups to the target system using *vgimport*. (Even though the device path for the newly added storage may be different from the path on the source system, *vgimport* will successfully import the volume group because of the information recorded in the map file by the *vgexport –s* option that was used on the source system. Without the *–s* option, we would have to manually edit the map file to specify the device path for the disks or LUNs we moved to the target system.) For example:

```
vgimport –s –m vg.map /dev/vgORA
```

For a raw-volume database, be sure to change the volume group's special files to be owned by user "oracle" and group "dba" and then set permissions. For example:

```
chown –R oracle:dba /dev/vgORA
chmod 644 /dev/vgORA/*
```

Now activate the newly imported volume group:

```
vgchange –a y /dev/vgORA
```

and, if the database is file-system-based, mount the file system:

```
mount /dev/vg_sample/lvol_db /opt/oracle/db
```

Logging back in as the oracle user, you should be able to start up the database instance without difficulties. You should now check for any natively compiled objects and recompile them for the target environment. (See the section entitled Recompiling natively compiled objects at the end of this paper.)

Multiple-instance Real Application Clusters (RAC)

This is a recommended procedure for migrating a Real Application Cluster database between a cluster of HP 9000 servers and an HP Integrity cluster in either direction – the instructions would apply equally in either case. We will thus refer to the two clusters as "source" and "target", and the two nodes in each cluster as "NodeSRC1", "NodeSRC2", "NodeTRGT1" and "NodeTRGT2". (For ease of documentation, we've made this example a two-node cluster, but the same procedure can be extended to a cluster of any number of nodes.)

We recommend that you read and understand the single-instance sample migration, above, prior to attempting a clustered migration. Our focus here will be mainly on the parts of the process which differ from that for the single-instance case.

The HP 9000 environment is assumed to be a two-node server running the HP-UX 11i v1 operating system (11.11). Note: Migration is only supported to an equivalent or better OS version, so if this HP 9000 is to be the target environment, it would have to be upgraded to at least the same HP-UX version running on the Integrity system. (See the note under the Migration prerequisites section above.)

The HP Integrity environment is assumed to be a two-node cluster running HP-UX 11i v2 (11.23) September 2004.

The source environment, regardless of whether it is the HP 9000 or HP Integrity cluster, has the following characteristics:

- Oracle Database 10gR2 (version 10.2.0.2) with the Real Application Clusters option
- Database implemented using raw LVM logical volumes (no file systems)
- Serviceguard eRAC (extension for RAC) installed to supply cluster membership

Prior to migration

If migrating from our HP Integrity cluster to the HP 9000 cluster, the OS on the HP 9000 must be upgraded from 11i v1 to at least 11i v2 September 2004 (11.23). To migrate from the HP 9000 to Integrity, the HP 9000 can remain on 11i v1.

For a 10gR2 migration such as this one, the prerequisites include the requirement that all OS patches specified in the Oracle 10gR2 installation guide for HP-UX be applied. Thus, the 10gR2 installation manual for the source environment should be consulted, and all required OS patches applied.

Our Oracle version, 10.2.0.2, meets the requirements for a 10gR2 migration.

Target-cluster preparation

The two target nodes should be set up in accordance with standard procedures for configuring and installing HP-UX, Serviceguard Extension for Real Application Clusters (SGeRAC), and Oracle Real Application Clusters. The Oracle RAC installation documentation for HP-UX (for HP 9000 or HP Integrity servers) should be followed, including the specifications for OS kernel parameters and root-volume file system sizes.

In the Oracle-home directory on the target system, set up the shell's environment file (.profile) to create the ORACLE_SID environment variable using the same SID as on the source system. (In the following examples, we will use environment-variable notation such as "$oh-src1" to refer to the Oracle home on source node 1 and "$oh-trgt1" for target node 1; you should substitute the actual paths when you type these commands.)

Using *rcp*, copy all the database configuration and parameter files (including the initSID.ora file) from each source node to the corresponding target node (assuming you are logged in to the target system):

```
rcp NodeSRC1:$oh-src1/dbs/*.ora $oh-trgt1/dbs/.
rcp NodeSRC1:$oh-src1/dbs/orapwsid $oh-trgt1/dbs/.
rcp -r NodeSRC1:$oh-src1/admin/sid* $oh-trgt1/admin/.
```

The last command shown above copies the admin directory, which contains bdump, udump, etc. Log in to the second target node and repeat the above commands to copy between source node 2 and target node 2.

Ensure that the access permissions and file owners for all these files and directories are the same on the target system as on the source system. (Improperly set file permissions would cause an ORA-01102 when opening the database.) Edit the files as appropriate for the target cluster.

Moving the database from the source cluster to the target cluster

In this example, we'll show how the database would be moved.

See the discussion under Alternative 2: Moving the database to the target system for single-instance databases, above, and, for each node in the source cluster, follow the procedures for shutting down the database instance and deactivating and exporting the volume group(s). (The map file can be generated on either source node and copied across the network to both target nodes). Follow the single-instance procedures for moving/recabling or reconfiguring the storage infrastructure to be accessible to the target environment, making sure that the storage infrastructure is visible to both target nodes, and sharable between them.

Once the storage infrastructure is visible to both target nodes, import the database volume group(s) to each of the nodes in the target cluster, following the instructions for moving a single-instance database above, specifically, the instructions for importing (vgimport) and changing ownership and permissions for the volume group special files (chown and chmod). To activate the volume group in shared mode, you will need to use the following vgchange command on both target nodes:

```
vgchange –a s /dev/vgrac
```

For ease of administration in a cluster environment, the usual practice is to create symbolic links which point to the device files identifying the raw logical volumes. We'll assume that these links were in use on the source cluster (and were located in a directory under the Oracle database home) and need to be set up on the target cluster as well. Assuming that the target's volume structure is the same as the source cluster, we need only copy the directory containing the links from the source cluster to the target. Log in to one of the source nodes as the oracle user, and use commands like the following to copy the links to each of the target nodes:

```
cd $oh-src1/oradata/racdb
rcp * NodeTRGT1:$oh-trgt2/oradata/racdb/*
rcp * NodeTRGT2:$oh-trgt2/oradata/racdb/*
```

Logging back in as the oracle user on each target node, you should be able to start up the database instance without difficulties. You should now check for any natively compiled objects and recompile them for the target environment.

Recompiling natively compiled objects

Oracle permits certain kinds of executable database objects to be compiled into the native object code of the host operating environment in order to improve performance when they are executed. Since HP 9000 and HP Integrity servers use different native object code formats, it is necessary to locate any database objects which were natively compiled on the source environment and recompile them after the database is migrated to the target environment. Failure to recompile a natively compiled object will result in an error similar to the following when the object is executed:

```
ORA-06549: PL/SQL: failed to dynamically open shared object (DLL):
'/XXXXX/native_plsql/NATIVE_1__SYSTEM__P__58827.so' is not a valid load
module: Bad machine type
```

To create a list of natively compiled PL/SQL objects on Oracle Database 10*g* Release 2, you can query the appropriate database views, similar to the following:

```
select NAME, PLSQL_CODE_TYPE, TYPE
from USER_PLSQL_OBJECT_SETTINGS
where
PLSQL_CODE_TYPE='NATIVE';
```

or the following:

```
select NAME, PLSQL_CODE_TYPE, TYPE
from DBA_PLSQL_OBJECT_SETTINGS
where
PLSQL_CODE_TYPE='NATIVE' and
and NAME like '%<XYZ>%';
```

To recompile a specific PL/SQL object, use the following sqlplus command:

```
alter <type> <name> compile;
```

Conclusion

HP co-developed the Intel Itanium architecture to be compatible with HP's PA-RISC architecture, and this has led to data migrations that are extremely simple, require no data conversions, and introduce no performance penalties for a migrated database.

HP customers can be confident in migrating their Oracle databases in either direction between the PA-RISC HP 9000 platform and the Intel Itanium-based HP Integrity platform – confident that their migration will go smoothly and confident that their database will perform optimally despite having been migrated from another architecture.

For more information

- Failover of Oracle Database between HP 9000 (PA-RISC) servers and HP Integrity (IPF) servers under HP-UX 11i

- Oracle "My Support" site: http://metalink.oracle.com (note: site is restricted to Oracle support customers). Specific note IDs:
  - 266220.1 (migrating an Oracle 9i database from HP 9000 to HP Integrity)
  - 395982.1 (Data Guard in mixed HP 9000/HP Integrity environments)
  - 402497.1 (migrating an Oracle 9i or 10g database between HP 9000 and HP Integrity)
  - 427712.1 (migrating an Oracle 10g or 11g database between HP 9000 and HP Integrity)
  - 549085.1 (restoring an RMAN database backup from HP-UX 11.11 on HP 9000 to HP-UX 11.31 on HP Integrity)

- PL/SQL Native Compilation (NCOMP) – FAQ: http://www.oracle.com/technology/tech/pl_sql/htdocs/ncomp_faq.html

- Compiling PL/SQL Code for Native Execution 10g Release 2 (10.2): http://download-west.oracle.com/docs/cd/B19306_01/appdev.102/b14261/tuning.htm#sthref2263

To help us improve our documents, please provide feedback at http://h20219.www2.hp.com/ActiveAnswers/us/en/solutions/technical_tools_feedback.html.

4AA1-7190ENW, Revision 2, July 2009