

Using HP Process Resource Manager with Oracle databases



Overview	3
References	3
Features	4
FSS PRM groups	4
pSet PRM groups	4
Database Resource Manager	5
Memory resource groups	5
Managing private memory	5
Managing shared memory.....	6
Secure Resource Partitions.....	6
Application support.....	6
HP Process Resource Manager and Oracle integration summary.....	7
Partial resource isolation	7
Advantages.....	7
Disadvantages	7
Total resource isolation	8
Advantages.....	8
Disadvantages	8
Configuring HP Process Resource Manager.....	9
Oracle and CPU (core) counts.....	10
Case study	10
Using the GUI to configure HP Process Resource Manager	10
Using the command-line interface to configure HP Process Resource Manager	13
Process management.....	14
Oracle process listing	15
Managing listener processes.....	16
Placing listener processes with HP Process Resource Manager C.03.01 and earlier	16
Placing listener processes with HP Process Resource Manager C.03.02 and later	17
Pattern matching rules.....	17
Validation.....	17
For more information.....	19

Overview

This paper describes two approaches for successfully configuring HP Process Resource Manager (PRM) to allow multiple instances of an Oracle® database to execute on the same system without any instance usurping the resources from another instance. The first approach guarantees a user-specified CPU and memory resource level to an Oracle database instance only when the system is at peak load. Otherwise, when the system is not at peak load, Oracle can consume additional CPU or memory resources if available. The second approach gives each Oracle database instance complete CPU and memory resource isolation. In this case, Oracle is not permitted to consume additional CPU or memory resources beyond its entitlement even if the system is not at peak load and the resources are available.

This paper also identifies the appropriate versions of PRM and Oracle and describes certain restrictions. The example used in this paper applies to a single system running multiple Oracle database instances. It does not apply to systems running Oracle Parallel Server.

In addition, this document covers issues specific to PRM and Oracle interoperability. It does not provide details on how to configure and use PRM for general-purpose operations. Also, details on the configuration and use of Oracle and its Database Resource Manager are not described. See the “References” section for additional information on using these products.

This document describes the two PRM features that are critical to the successful blending of PRM and Oracle databases: processor set (pSet) support and Memory Resource Groups (MRGs) for both private memory and shared memory. In addition, this paper discusses the use of the Oracle Database Resource Manager that is available starting with Oracle 8i. Next, it describes recommended configuration procedures. The paper ends with a case study that gives an example of a system running three Oracle database instances, with each having different CPU and memory resource requirements.

References

For more details on configuring and using HP PRM, see the HP Process Resource Manager User’s Guide at <http://docs.hp.com/en/ha.html#Process%20Resource%20Manager>.

For a good introduction to PRM, see the Process Resource Manager Overview at <http://h20338.www2.hp.com/hpux11i/downloads/prm.overview.pdf>.

For more information on configuring and using Oracle’s Database Resource Manager, see http://www.oracle.com/technology/products/manageability/database/pdf/twp03/twp_oracle%20database%2010q%20resource%20manager.pdf.

For a benchmark paper showing Oracle with PRM, see “Benchmarking Oracle Database with PRM shared memory management” at <http://h20338.www2.hp.com/hpux11i/downloads/PRM.Oracle.SharedMemory.pdf>

Features

The following features should be understood before Oracle can be successfully integrated with PRM. These features are described in detail in the sections that follow.

- PRM groups. PRM enables you to separate processes into Fair Share Scheduler (FSS) PRM groups and into PRM groups based on pSets. With FSS PRM groups, the FSS allocates CPU resources to certain processes. In addition, PRM can create pSets, with the processes getting CPU resources allocated based on the number of CPUs, or cores, in the pSet. (A core is the actual data processing engine within a processor. A single processor might have multiple cores. Starting with HP-UX 11i v3, a core can support multiple execution threads. This feature is known as Hyper-Threading. With Hyper-Threading disabled or unavailable, each core is seen as a CPU. With Hyper-Threading enabled, each core can be seen as multiple, logical CPUs.)
- Oracle Database Resource Manager. Introduced in Oracle 8i, the Database Resource Manager is an optional feature that can be successfully used with PRM C.02.00 or later when PRM is configured appropriately.
- MRGs. The MRG feature introduced with PRM C.02.00 on HP UX 11i plays an important role in isolating an Oracle database instance. When used in conjunction with pSet PRM groups, both private memory and CPU resources can be isolated for a specific Oracle database instance. PRM C.03.01 on HP-UX 11i v2 Update 2 or later introduced MRGs that can isolate shared memory, which is useful for an Oracle database's System Global Area (SGA).
- Secure Resource Partitions. HP-UX 11i v2 (B.11.23) offers the Security Containment feature. This feature provides file and process isolation in secure compartments. Combining Security Containment and PRM, you can form Secure Resource Partitions in which PRM controls resource allocation for your secure compartments. Placing Oracle database instances in their own Secure Resource Partitions enables you to provide another level of isolation to the instances.
- Application support. PRM application support provides the ability to place processes in specific PRM groups.

FSS PRM groups

Each FSS PRM group is assigned a user-specified number of shares of the CPU resource pool. The CPU resource pool consists of all the cores not assigned to a pSet. It is dedicated to the FSS PRM groups. The shares value of a FSS PRM group is translated internally to a percentage of total system CPU cycles. However, that percentage changes as other FSS PRM groups are added or deleted because the shares ratio (group shares/total shares) changes. All FSS PRM groups must be associated with, and share the resources of, the CPU resource pool. Although the CPU resource pool initially consists of all the cores from the system, that number decreases as cores are assigned to pSet PRM groups.

pSet PRM groups

Defining a pSet PRM group creates a pSet on the system. The pSet PRM group enables users and applications assigned to this group to run within this pSet. You can associate only one pSet PRM group with each pSet.

To illustrate how PRM works with pSets, consider an eight-core system with four cores assigned to a pSet PRM group. The pSet PRM group has a hard limit of 50% of the system CPU resources. Other FSS PRM groups can still be created, but their shares allocations now come from the remaining unassigned cores in the CPU resource pool.

The PRM pSets feature is supported on HP-UX 11i running PRM C.02.00 or later.

Database Resource Manager

Oracle 8i and later have their own resource management tool named the Database Resource Manager. The Oracle Database Resource Manager enables the database administrator to prioritize the allocation of CPU resources among various groups of users and applications on a per-session basis. The Database Resource Manager can specify CPU resource percentages, execution limits, degree of parallelism, active session pool, and size of the undo pool for specific users and applications. Hierarchies are supported as well.

Some of the Database Resource Manager allocation algorithms could operate inconsistently with the manner in which PRM allocates resources if the Oracle database instance processes are assigned to an FSS PRM group. When the system is at peak load, that FSS PRM group is guaranteed a percentage of the CPU resources. However, if the Database Resource Manager is running, it assumes that all the CPUs it detected at instance startup are still available. The Database Resource Manager is not aware if the PRM FSS attempts to restrict it to a user-specified fraction of the CPU resources. For example, even though PRM allocates 50% of the CPU cycles to an Oracle database instance on a four-core system, the Database Resource Manager assumes it has access to all four cores, or 100% of the CPU cycles, and will allocate CPU resources to users and applications based on that assumption. It is not aware that PRM is allowing it to use only 50% of the CPU cycles. Because of this, the Database Resource Manager can produce unexpected results in terms of resource allocation and user access priorities. For this reason, the use of the Database Resource Manager is not recommended for Oracle instances that are assigned to FSS PRM groups.

However, when the Oracle database instance is placed in a pSet PRM group, there is no inconsistency. The Database Resource Manager determines how many cores are available to it at instance startup and continues to monitor this value so it can dynamically adjust its internal algorithms as cores are added and removed from its operating environment (that is, the pSet).

Memory resource groups

MRGs enable you to logically partition private (that is, user) memory and shared memory. (Management of shared memory is available starting with PRM C.03.01 on HP-UX 11i v2 Update 2 and later.) MRGs do not, however, support the partitioning of user memory resource types such as shared text or shared library code.

Managing private memory

You can associate memory resource entitlements with both FSS and pSet PRM groups. This entitlement represents a guaranteed minimum amount of private real memory that the processes within a PRM group are allowed to consume at times of memory contention. The total memory demands from the applications running on a system can fluctuate and vary significantly as time progresses. When there is no memory contention, PRM groups can freely borrow memory from other PRM groups as needed and exceed their memory entitlements to satisfy their needs at the moment. However, the borrowed memory must be returned when the system next experiences memory contention.

With the introduction of MRGs on HP-UX 11i, PRM C.02.00 and later offer memory isolation. Memory isolation can be applied on a per-group basis to prevent the free flow of memory resources among PRM groups when the system is not experiencing memory contention. Memory cannot be loaned out to or borrowed from other PRM groups. Memory isolation is useful for applications such as Oracle that need dedicated memory resources and for applications that tune their own memory needs based on their allocated resources.

Managing shared memory

By default, all shared memory is allocated from the PRM_SYS group.

Starting with HP-UX 11i v2 Update 2 and PRM C.03.01, PRM can control shared memory allocations on a PRM group basis. You can selectively specify shared memory management for the groups that need it—omitting groups where shared memory management would not be helpful.

You set a minimum allocation in megabytes for a group's shared memory. (The size you should allocate is usually available from the configuration settings for the consuming application, as is the case with the Oracle SGA size.)

Capping is not available for shared memory.

Secure Resource Partitions

The HP-UX Security Containment feature is available starting with 11i v2 (B.11.23) and provides secure compartments. These compartments provide isolation between unrelated resources to prevent damage to a whole system if a compartment is penetrated. Applications configured in compartments have restricted access to resources outside their configured compartments.

You can place one or more secure compartments in a single PRM group to manage the resource allocation for those secure compartments. This combination is known as Secure Resource Partitions. With this combination, you can securely combine Oracle instances on a single system and still ensure they get the resources they need.

Application support

PRM's application support enables you to put processes that are not fully qualified executables (that is, processes created by other processes) into separate PRM groups. For the Oracle database administrator, this means the ability to allocate different CPU and memory resources to Oracle database instances by modifying the PRM configuration. The PRM application manager periodically checks the system for new processes. Those processes matching PRM application records are placed in the configured PRM groups.

The PRM partitioning is done between separate Oracle database instances—not within a single instance. That is, all Oracle processes related to a specific database instance must be assigned to the same PRM group, including Oracle server processes (shadow processes) that run on behalf of users accessing the database.

`$ORACLE_HOME` and `$ORACLE_SID` are standard environment variable names for the Oracle home directory and the name of the Oracle database instance, respectively. When you open an Oracle database instance, the `$ORACLE_HOME/bin/oracle` executable renames itself using the UNIX® environment variable `$ORACLE_SID` for a given database. The `$ORACLE_SID` environment variable is used in generating Oracle process names: `ora_<process_name>_<ORACLE_SID>`. For example, all Oracle processes associated with the Sales database, where `$ORACLE_SID` is Sales, would be renamed with Sales appended to the process name.

HP Process Resource Manager and Oracle integration summary

The approaches for using PRM with Oracle are summarized in the following two sections. These approaches are:

- Partial resource isolation—Isolating CPU and private memory resources only when the system is at peak load but allowing CPU and private memory resources to be shared when the system is not at peak load
- Total resource isolation—Complete isolation of both the CPU and the private memory resources

With either form of isolation, you can also specify a minimum amount of per-group shared memory.

If you plan on using the Oracle Database Resource Manager, you must use total resource isolation. As explained in previous sections, the Database Resource Manager and PRM must agree on the amount of CPU and memory resources available.

Partial resource isolation

Partial resource isolation means that the Oracle database instances are assigned to FSS PRM groups and are all executing within a given set of cores. The CPU cycles and private memory used by one database instance can be loaned out to database instances running in other FSS PRM groups when the database instance is lightly loaded. Likewise, when the Oracle database instance is heavily loaded, it can borrow CPU cycles and private memory from other FSS PRM groups that are not under stress. On HP-UX 11i v2 and later, you can dedicate a minimum amount of shared memory to an FSS PRM group. This memory can be assigned on a group-by-group basis. On HP-UX 11i v3 and later, you can set an upper bound for CPU resource consumption by an FSS PRM group on a group-by-group basis.

Advantages

The advantages of partial resource isolation are:

- More efficient use of CPU and private memory resources because not all Oracle database instances are likely to be under heavy load constantly. Any easing of load on one of the database instances allows other instances to grab those resources.
- Significantly improved performance for the heavily stressed database instance when one or more of the other instances are lightly loaded. This advantage results from the FSS PRM group of the heavily stressed instance borrowing resources from the other FSS PRM groups containing lightly loaded database instances.
- A fraction of a core can be allocated to a FSS PRM group for more fine-grained tuning.

Disadvantages

The disadvantages of partial resource isolation are:

- The Oracle Database Resource Manager is not supported.
- Less deterministic behavior because the amounts of CPU and private memory resources are dependent on the loads on the other database instances. These loads on other database instances can vary significantly and are unpredictable.

Total resource isolation

Total resource isolation means that a single Oracle database instance runs in a PRM group that has access to a fixed pool of cores and private memory. The PRM group cannot loan these resources to other PRM groups, nor can it borrow additional resources from other PRM groups. If cores are not used, they go idle. If memory is not used, it is wasted. However, both cores and memory remain dedicated to the Oracle instance running in the associated PRM group.

To achieve total resource isolation, you:

- Assign the Oracle database instance to a pSet PRM group, providing isolation of the CPU resource
- Enable the MRG private memory isolation feature, providing isolation of the memory resource

PRM's application manager, when combined with the MRG private memory isolation and pSet features of PRM, can be used to completely isolate and manage resource allocation for multiple instances of Oracle databases. The Oracle Database Resource Manager can then be used to independently allocate resources to applications and users of an Oracle instance within the context of a pSet.

NOTE

You do not have to use private memory isolation when using CPU resource isolation. With private memory isolation, your database instance is not affected by the memory use of any other workload on the system. However, it also cannot borrow unused resources from other workloads. Without private memory isolation, your instance can take advantage of unused memory when doing a large sort or select. However, performance will be inconsistent because the extra memory might not always be available.

You can dedicate a minimum amount of shared memory to a pSet PRM group—on a group-by-group basis. This feature enables you to manage an instance's SGA.

Advantages

The advantages of total resource isolation are:

- Fixed and unchanging CPU and memory resource entitlements allow for more deterministic behavior because surplus CPU and memory resources are not available and any loaned-out resources do not have to be reclaimed when needed.
- Support for the Oracle Database Resource Manager because PRM and the Database Resource Manager are in agreement on what CPU and memory resources are available.
- Support for Oracle 9i (and later) Program Global Area (PGA) memory management. When long queries are run, Oracle processes can relinquish memory during or after a query so that other processes within the same instance can use this memory. Without memory isolation, there is no way to guarantee that other processes in the same instance will use this newly freed memory.

Disadvantages

The disadvantages of total resource isolation are:

- Resources are wasted because unused memory and CPU cycles go idle or are not used.
- The CPU resource can only be allocated in course-grained chunks based on the core. A fraction or percentage of a core cannot be allocated to a pSet PRM group.

Configuring HP Process Resource Manager

The following procedure outlines step-by-step instructions for configuring PRM to work with Oracle. Steps involving the startup and configuration of the Oracle database are outside the scope of this document. See the appropriate Oracle documentation. Also, see the HP Process Resource Manager User's Guide for more detailed configuration information.

You can configure PRM using its interface in HP System Management Homepage or on the command line (using `prmconfig`). The following steps have been generalized so that they apply to both approaches. The case study in the following section gives the specifics for a GUI approach and an associated PRM configuration file that can be created in an editor and loaded using `prmconfig` to give the same result.

NOTE

If you are configuring PRM's shared memory management, load the PRM configuration before starting the Oracle database. If the database is already running when you load the PRM configuration, restart the database.

1. Verify the versions of HP-UX, Oracle, and PRM that are installed on your system. This paper was written based on functionality provided by HP-UX 11i v2 Update 2, Oracle 8i, Oracle 9i, Oracle 10g, and PRM C.03.02 or later.
2. Define your PRM groups. Review the advantages and disadvantages of FSS PRM groups compared to pSet PRM groups as described in the "HP Process Resource Manager and Oracle integration summary" section on page 7, and decide what type of PRM groups are appropriate for your situation.
3. Define your MRGs.
4. Decide whether you want private memory entitlements for your Oracle database instances. (If you assign a private memory entitlement to one PRM group, you must assign them to all groups.)
5. Review the section "Managing private memory" on page 5 to determine if you need memory resource isolation for your Oracle instances. If so, enable memory isolation for the appropriate PRM groups.
6. Decide whether you want shared memory management for the instance's SGA. You can manage shared memory for just one PRM group, multiple PRM groups, or all PRM groups.
7. Define your user records to reference the new PRM group definitions.
8. Add application record definitions for each Oracle database instance. Enter the full name of the Oracle executable as the application name, the name of the PRM group, and the alternate name for the Oracle application. The alternate name must have the format `ora* <$ORACLE_SID>` to match all processes related to an individual Oracle database instance.
9. Place each listener process in its appropriate PRM group. This step must be completed manually by using the `prmmove` command or by placing the commands in an executable script. See "Managing listener processes" on page 16 for additional information.
10. (Available starting with HP-UX 11i v2) Create Secure Resource Partitions, if desired, using the `prm2scomp` utility to generate a Security Containment configuration based on the PRM configuration. This utility also modifies your PRM configuration to map the compartments to your PRM groups. Load and activate the Security Containment configuration using the `setrules` utility.
11. Load the PRM configuration and enable the PRM application manager and the memory manager.

Oracle and CPU (core) counts

Oracle only checks the CPU count (core count) at startup. If an instance is moved to another pSet PRM group after this check is made, Oracle does not take into account that the new pSet might have a different core count. Therefore, ensure that the parent process for your database processes is in the correct PRM group (and not the OTHERS group) when it starts the database processes. Options for accomplishing this placement include:

- Using `prmrn` to start the parent process in the correct PRM group
- Performing the following steps:
 1. Invoke `sqlplus` and connect to the database.
 2. Wait at least one PRM application manager interval.

By default, this interval is 30 seconds. Verify your interval using the following commands:

```
# /opt/prm/bin/prmconfig | grep interval
```

3. Start your database instances.

- Creating and using a wrapper script of the form:

```
#!/bin/sh
prmmove -p $$ target PRM group
<start your database instances>
```

This script uses the `prmmove` command to place itself in the desired `target_PRM_group`. Any processes the script starts then inherit its PRM group. So, when the script starts the database instances, those instances start in the proper PRM group.

Case study

The following case study examines a system containing eight cores and running three Oracle database instances:

- The Sales database is used for critical order processing and requires 50% of the total system's CPU cycles and should be totally isolated from other Oracle instances.
- The Support database is critical for customer service but needs fewer CPU cycles (about 25% of the total system's CPU cycles) and should also be totally isolated from other Oracle instances.
- The Mktg database, used for analysis and planning, has a lower priority than the others and requires about 25% of the total system's CPU cycles. It does not require total resource isolation and can share its resources with other non-database applications when the system is lightly stressed.

The required configuration is defined first using the graphical user interface (GUI). Then, the resulting PRM configuration file is given. Then this configuration is loaded using the `prmconfig` command. Next, general information on the management of the processes is provided.

Using the GUI to configure HP Process Resource Manager

The following PRM configuration was created to allocate the appropriate amount of system resources to each of the three database instances. The screens in this section are captured from the PRM C.03.02 interface in HP System Management Homepage (SMH). This interface is available with PRM C.03.02 and later. PRM C.03.03.01 provides an interface in HP Systems Insight Manager in addition to the HP SMH interface.

Figure 1 shows the PRM group definitions and either their CPU resource shares for FSS PRM groups or allocation of cores for pSet PRM groups.

Figure 1. PRM group and CPU configuration

PRM Configuration File Name: /etc/opt/prm/conf/oracle.prm

Existing PRM Groups

New PRM Group Definition:

Enter the PRM group name: Select the PRM group type: FSS PSET HIER

Enter number of shares for this FSS group: Enter the CPU Cap % Value: Enter number of cores for this PSET group: Enter the Hyper-Threading status: Default

Total Cores: 4, Available Cores: 0 (1 core reserved for FSS groups)

refresh Add PRM Group Record

<input type="checkbox"/>	PRM Group	Group Type	Shares	Percentage	Capping %	Number of Cores	LCPU
<input type="checkbox"/>	Mktg	FSS	100	12.5%	<input type="text"/>	<input type="text"/>	Default
<input type="checkbox"/>	OTHERS	FSS	100	12.5%	<input type="text"/>	<input type="text"/>	Default
<input type="checkbox"/>	Sales	PSET	<input type="text"/>	50.0%	<input type="text"/>	2	Default
<input type="checkbox"/>	Support	PSET	<input type="text"/>	25.0%	<input type="text"/>	1	Default

The Mktg and OTHERS PRM groups are FSS PRM groups because they have a shares allocation. The Sales and Support PRM groups are pSet PRM groups because they have their number of cores specified. Each group's CPU resource percentage is based on the total system CPU resources.

Figure 2 shows the memory configuration.

Figure 2. PRM memory configuration

PRM Configuration File Name: /etc/opt/prm/conf/oracle.prm

Existing Memory Records

Available Memory: 6271 Mbytes

refresh

PRM Group	Shares	Private MB	Cap (%)	Isolated	Shared MB
Mktg	100	1253.6 MB	<input type="text"/>	false	<input type="text"/>
OTHERS	100	1253.8 MB	<input type="text"/>	false	<input type="text"/>
Sales	200	2507.2 MB	<input type="text"/>	false	<input type="text"/>
Support	100	1253.6 MB	<input type="text"/>	false	<input type="text"/>

Each PRM group has a shares allocation for private memory. This allocation is based on the total system memory available. None of the groups has memory isolation enabled. Thus, they can borrow or lend memory resources from each other.

Figure 3 shows the PRM user configuration with the Oracle user having access to both the FSS PRM group Mktg and the pSet PRM groups Sales and Support.

Figure 3. PRM user configuration

The screenshot shows the PRM configuration interface with the 'Users' tab selected. The PRM Configuration File Name is /etc/opt/prm/conf/oracle.prm. Under 'Existing User Records', there is a 'New PRM User Definition' section with input fields for user name, primary PRM group (set to 'Mktg'), and additional PRM groups. Below this is a table of existing records:

<input type="checkbox"/>	User Name	Primary PRM Group	Additional PRM Groups
<input type="checkbox"/>	bmeade	OTHERS	
<input type="checkbox"/>	jstark	OTHERS	
<input type="checkbox"/>	klane	OTHERS	
<input type="checkbox"/>	oracle	Mktg	Sales,Support

If users are not logging in to HP-UX directly, records for each individual user are not required.

If users are logging in to HP-UX and accessing interactive applications, the users' shells and applications should execute in PRM groups other than those used by the Oracle server processes (OTHERS in this example).

PRM's application records enable you to separate the instance processes into individual PRM groups. The Oracle database executable is the same in every case, but the alternate name specifies a pattern to match based on the instance name, \$ORACLE_SID. The application manager recognizes the appropriate processes based on the supplied pattern and assigns them to the specified PRM groups.

So the application is /u01/app/oracle/product/10.2.0/db_1/bin/oracle.

To place processes in the Sales PRM group, use an alternate name of ora*Sales. Processes starting with "ora" and ending with "Sales" are put in the Sales PRM group. This pattern includes Oracle database server processes and shadow processes.

Similarly, use an alternate name of ora*Support to place processes in the Support group and an alternate name of ora*Mktg to place processes in the Mktg group.

Create a record to place all other child processes associated with starting an Oracle database instance in the PRM group OTHERS.

The application manager checks all running processes against the PRM configuration application records, moves the processes pertaining to each Oracle database instance to the appropriate PRM groups, and writes a record to the /var/adm/syslog/syslog.log file.

Using the command-line interface to configure HP Process Resource Manager

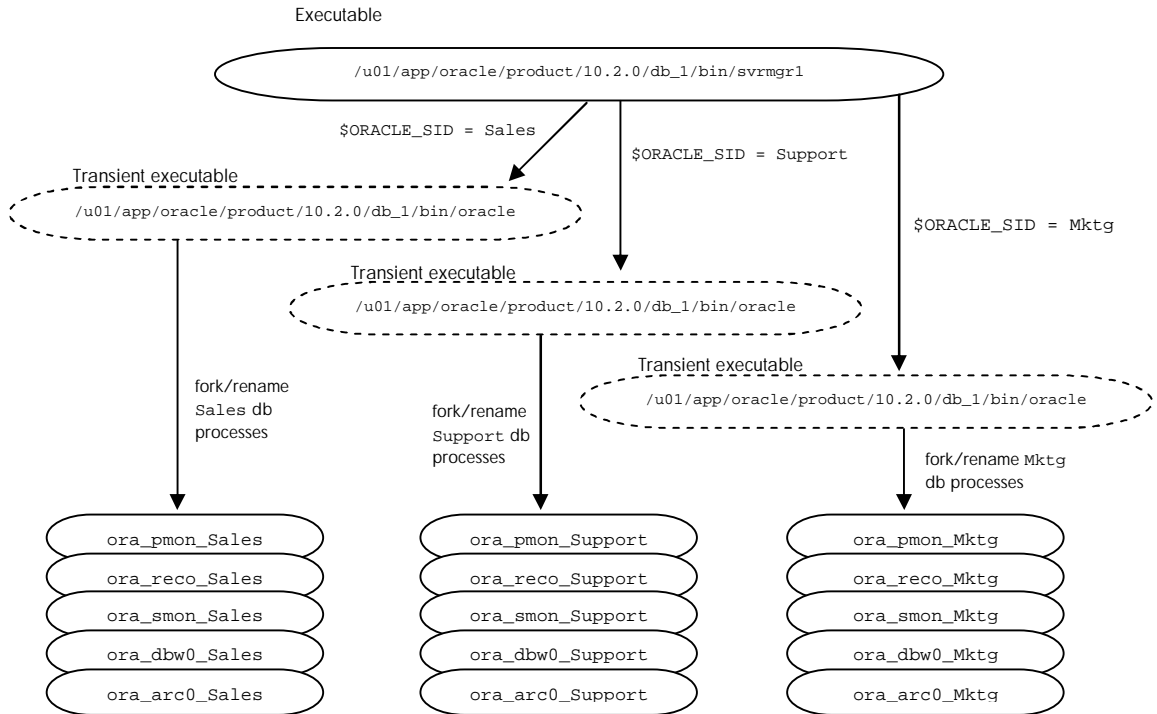
In addition to the SMH interface and the SIM interface, you can manually create the PRM configuration using an editor. You activate the configuration with the `prmconfig` utility. The configuration defined in the previous sections by the GUI appears in the following PRM configuration file.

```
#
# Group/CPU records
#
Mktg:2:100::
OTHERS:1:100::
Sales:PSET:::4:
Support:PSET:::2:
#
# Memory records
#
#!PRM_MEM:2:100::::
#!PRM_MEM:1:100::::
#!PRM_MEM:Sales:200::::0:0:
#!PRM_MEM:Support:100::::0:0:
#
# Application records
#
/u01/app/oracle/product/10.2.0/db_1/bin/oracle::::Sales,ora*Sales
/u01/app/oracle/product/10.2.0/db_1/bin/oracle::::Support,ora*Support
/u01/app/oracle/product/10.2.0/db_1/bin/oracle::::Mktg,ora*Mktg
/u01/app/oracle/product/10.2.0/db_1/bin/oracle::::OTHERS
#
# User records
#
bmeade::::OTHERS
jstark::::OTHERS
klane::::OTHERS
oracle::::OTHERS,Mktg,Sales,Support
```

Process management

Figure 4 shows that each database instance (Sales, Support, and Mktg) is launched from a unique instance of the same executable: `/u01/app/oracle/product/10.2.0/db_1/bin/oracle`. The processes for the database instance are renamed according to the variable `$ORACLE_SID`. This is important because the fully qualified executable in the PRM application record must be the one that launches the processes you want to place in a specific PRM group.

Figure 4. Oracle processes



Oracle process listing

The `ps -ef` output looks similar to the following output when all three database instances are open:

```
oracle 1832      1  0 09:35:48 ?          0:00 ora_reco_Mktg
oracle 1822      1  3 09:35:48 ?          0:00 ora_dbw0_Mktg
oracle 1873      1  0 09:35:55 ?          0:00 ora_smon_Sales
oracle 1830      1  0 09:35:48 ?          0:00 ora_smon_Mktg
oracle 1862      1  4 09:35:55 ?          0:00 ora_pmon_Sales
oracle 1834      1  0 09:35:49 ?          0:00 ora_arc0_Mktg
oracle 2116    2115  0 09:49:08 ?          0:00 oracleMktg
(DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq)))
oracle 1905      1  4 09:36:01 ?          0:00 ora_pmon_Support
oracle 1864      1  3 09:35:55 ?          0:00 ora_dbw0_Sales
oracle 1877      1  0 09:35:55 ?          0:00 ora_arc0_Sales
oracle 1769      1  0 09:35:02 ?          0:00
/u01/app/oracle/product/10.2.0/db_1/bin/tnslsnr LISTENER_Sales
oracle 1875      1  0 09:35:55 ?          0:00 ora_reco_Sales
oracle 1907      1  4 09:36:01 ?          0:00 ora_dbw0_Support
oracle 1819      1  4 09:35:48 ?          0:00 ora_pmon_Mktg
oracle 1779      1  0 09:35:23 ?          0:00
/u01/app/oracle/product/10.2.0/db_1/bin/tnslsnr LISTENER_Mktg -
oracle 1774      1  0 09:35:12 ?          0:00
/u01/app/oracle/product/10.2.0/db_1/bin/tnslsnr LISTENER_Support
oracle 1916      1  0 09:36:01 ?          0:00 ora_smon_Support
oracle 1918      1  0 09:36:01 ?          0:00 ora_reco_Support
oracle 1920      1  0 09:36:01 ?          0:00 ora_arc0_Support
jstark 1959    1958  0 09:41:36 ?          0:00 oracleSales
(DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq)))
bmeade 2031    2030  0 09:46:36 ?          0:00 oracleSupport
(DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq
klane  2124    2123  0 09:50:03 ?          0:00 oracleMktg
(DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq)))
```

This process list also shows Oracle shadow processes for users connected to the database instances. The shadow processes have the process IDs of 1959, 2031, and 2124. Shadow processes are created when users connect to a database instance. In this example, three users are connected to different database instances. User `jstark` is connected to the Sales database, user `bmeade` is connected to the Support database, and user `klane` is connected to the Mktg database. The shadow processes relay database requests from users to the Oracle database. The process names for shadow processes have the same pattern as the Oracle database server processes. Consequently, these processes are caught by the application record pattern and moved to the appropriate PRM groups.

This process list also shows Oracle listener processes. Although you can configure Oracle to have a single listener process for all database instances, it is preferable to have a separate listener process for each database instance so that the listener process can be moved to the same PRM group as its associated database instance. The listener processes in this list have process IDs of 1769, 1779, and 1774.

Managing listener processes

Although listener processes can be associated with a particular database instance, they are actually started as part of the networking processes that run with Oracle. In `ps` output, they appear as fully qualified executables:

```
oracle 1769      1  0 09:35:02 ?          0:00
/u01/app/oracle/product/10.2.0/db_1/bin/tnslsnr LISTENER_Sales
oracle 1779      1  0 09:35:23 ?          0:00
/u01/app/oracle/product/10.2.0/db_1/bin/tnslsnr LISTENER_Mktg -
oracle 1774      1  0 09:35:12 ?          0:00
/u01/app/oracle/product/10.2.0/db_1/bin/tnslsnr LISTENER_Suppor
```

The three listeners have exactly the same executable name, but with different parameters.

Placing listener processes with HP Process Resource Manager C.03.01 and earlier

Using PRM C.03.01 and earlier, the listener process associated with the Sales database cannot be placed in the Sales PRM group by using an application record in the PRM configuration file. To put listener processes in the same PRM group as the database instance, use the following `prmmove` command with the listener's process ID:

```
# prmmove Sales -p 1769
```

You might want to automate this procedure with a script that runs after all databases are online:

```
#!/usr/bin/sh
# Script to move Oracle listener processes to PRM groups
# Find listener process for Sales database
pid1= `ps -ef | grep tnslnr | grep LISTENER_Sales | \
awk '{print $2}'`
# Move listener process for Sales database
prmmove Sales -p $pid1
# Find listener process for Support database
pid2 = `ps -ef | grep tnslnr | grep LISTENER_Supp | \
awk '{print $2}'`
# Move listener process for Support database
prmmove Support -p $pid2
# Find listener process for Mktg database
pid3 = `ps -ef | grep tnslnr | grep LISTENER_Mktg | \
awk '{print $2}'`
# Move listener process for Mktg database
prmmove Mktg -p $pid3
```

This script does not handle the case in which a database instance has multiple listener processes.

Placing listener processes with HP Process Resource Manager C.03.02 and later

Starting with PRM C.03.02, PRM supports Extended Regular Expressions in alternate names. This support enables you to create application records for listener processes.

To put the listener processes in the same PRM groups as their database instances, use the following application records:

```
/u01/app/oracle/product/10.2.0/db_1/bin/tnslsnr::::Sales, 'tnslsnr.*Sales'  
/u01/app/oracle/product/10.2.0/db_1/bin/tnslsnr::::Mktg, 'tnslsnr.*Mktg'  
/u01/app/oracle/product/10.2.0/db_1/bin/tnslsnr::::Support,  
'tnslsnr.*Support'
```

Pattern matching rules

Pattern matching follows the rules for Basic Regular Expressions and, with PRM C.03.02 and later, Extended Regular Expressions, as outlined in the regexp(5) manpage.

Because pattern matching is intended to match a collection of alternate names to a fully qualified executable, PRM was engineered to ensure that other applications and processes matching the pattern were not inadvertently put in the wrong PRM group. PRM does this by comparing file system inode numbers. For a process matching the alternate name pattern to be placed in the configured PRM group, it must share the file system inode number of the parent process.

The Oracle database administrator does not have to worry about inode numbers because all processes launched from the same executable (for example, Oracle database instances) share the same file system inode number.

Validation

Use the `ps -efP` command to validate that all database processes are running in the correct PRM groups. The column labeled PRMID indicates each process's PRM group.

UID	PRMID	PID	PPID	C	STIME	TTY	TIME	COMMAND
oracle	Mktg	1832	1	0	09:35:48	?	0:00	ora_reco_Mktg
oracle	Mktg	1822	1	4	09:35:48	?	0:00	ora_dbw0_Mktg
oracle	Sales	1873	1	0	09:35:55	?	0:00	ora_smon_Sales
oracle	Mktg	1830	1	0	09:35:48	?	0:00	ora_smon_Mktg
oracle	Sales	1862	1	2	09:35:55	?	0:00	ora_pmon_Sales
oracle	Mktg	1834	1	0	09:35:49	?	0:00	ora_arc0_Mktg
oracle	Mktg	2116	2115	0	09:49:08	?	0:00	oracleMktg (DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq)))
oracle	Support		1905	1	2	09:36:01	?	0:00 ora_pmon_Support
oracle	Sales	1864	1	1	09:35:55	?	0:00	ora_dbw0_Sales
oracle	Sales	1877	1	0	09:35:55	?	0:00	ora_arc0_Sales
oracle	Sales	1769	1	0	09:35:02	?	0:00	oracle
/u01/app/oracle/product/10.2.0/db_1/bin/tnslsnr LISTENER_Sales								

oracle	Sales	1875	1	0	09:35:55	?	0:00
ora_reco_Sales							
oracle	Support	1907	1	2	09:36:01	?	0:00
ora_dbw0_Support							
oracle	Mktg	1819	1	2	09:35:48	?	0:00
ora_pmon_Mktg							
oracle	Mktg	1779	1	0	09:35:23	?	0:00
/u01/app/oracle/product/10.2.0/db_1/bin/tnslsnr LISTENER_Mktg -							
oracle	Support	1774	1	0	09:35:12	?	0:00
/u01/app/oracle/product/10.2.0/db_1/bin/tnslsnr LISTENER_Support							
oracle	Support	1916	1	0	09:36:01	?	0:00
ora_smon_Support							
oracle	Support	1918	1	0	09:36:01	?	0:00
ora_reco_Support							
oracle	Support	1920	1	0	09:36:01	?	0:00
ora_arc0_Support							
jstark	Sales	1959	1958	0	09:41:36	?	0:00
oracleSales (DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq))							
bmeade	Support	2031	2030	0	09:46:36	?	0:00
oracleSupport (DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq							
oracle	OTHERS	2115	1183	0	09:49:05	ttyp1	0:00
sqlplus							
klane	Mktg	2124	2123	0	09:50:03	?	0:00
oracleMktg (DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq))							

For more information

For more information on HP Process Resource Manager, contact any of the HP worldwide sales offices or see the HP website at <http://www.hp.com/go/prm>.

The following references also provide useful information:

- HP Process Resource Manager User's Guide: <http://www.docs.hp.com/hpux/ha/>
- HP-UX 11i v2 Security Containment: <http://www.hp.com/go/securitycontainment>

© 2005, 2006, 2007 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Oracle is a registered US trademark of Oracle Corporation, Redwood City, California. UNIX is a registered trademark of The Open Group.

June 2007

