46
DBMS_CACHEUTIL

The DBMS_CACHEUTIL package provides an interface to control object-level cache fusion locking policies in an Oracle Real Application Clusters (Oracle RAC) environment and to manage assignments to the KEEP buffer pool for Oracle True Cache.

DBMS_CACHEUTIL Overview

DBMS_CACHEUTIL provides a collection of procedures related to Oracle Real Application Clusters (Oracle RAC) and Oracle True Cache.

With DBMS_CACHEUTIL, you can do the following:

- Control object-level cache fusion locking policies, such as object affinity, read-mostly, and object down-convert in an Oracle RAC environment.

  For example, you can fine-tune cache fusion locking policies for a specific configuration or workload at the object level. This might reduce locking overhead in an Oracle RAC environment and improve performance.

  - Affinity specifies an implicit, exclusive lock for an object to an instance, so that read accesses and updates to the object on this instance become more optimized. However, updates to the object on other instances become less optimized.
  - Read-mostly specifies an implicit, shared lock for an object in the cluster, so that read access to the object becomes more optimized. However, updates to the object on any instances become less optimized.
  - A down-convert converts an object's exclusive locks on an instance to shared mode for all instances in an Oracle RAC environment.

  > **Note:**
  >
  > Object affinity and read-mostly locking optimization works only for the primary RAC database, and it doesn't work for Oracle Active Data Guard.

- Manage object assignments to the KEEP buffer cache for an Oracle True Cache instance. True Cache is an in-memory, consistent, and automatically managed SQL and key-value (object or JSON) cache for Oracle Database.

You can apply the procedures in this package to tables, indexes, lobs, partitions (for partitioned objects), and subpartitions (for composite partitioned objects).

For non-partitioned objects, the input (schema and obj corresponds to OWNER and OBJECT_NAME in DBA_OBJECTS and should uniquely define a segment. That means it should have, for example, a valid DATA_OBJECT_ID in DBA_OBJECTS.

For partitioned or composite partitioned objects, the input (schema, obj, and partition) corresponds to OWNER, OBJECT_NAME, and SUBOBJECT_NAME in DBA_OBJECTS and should uniquely define a segment. That means it should have a valid DATA_OBJECT_ID in DBA_OBJECTS.

Except for depended indexes, this procedure does not apply for the dependent object segments, such as lob and partition or subpartition segments. It should be explicitly called on the segment if necessary.

Related Topics

- Real Application Clusters Administration and Deployment Guide

Related Topics

- Oracle True Cache User's Guide

DBMS_CACHEUTIL Security Model

The security model describes the privileges needed for using the `DBMS_CACHEUTIL` package.

Use the following for this package:

```
CREATE OR REPLACE PUBLIC SYNONYM dbms_cacheutil FOR sys.dbms_cacheutil


GRANT EXECUTE ON dbms_cacheutil TO dba
```

Summary of DBMS_CACHEUTIL Subprograms

This table lists the package subprograms in alphabetical order.

Table 46-1 DBMS_CACHEUTIL Package Subprograms

Subprogram	Description
DISSOLVE_AFFINITY Procedure	This procedure dissolves the object affinity from a node in an Oracle RAC environment.
DISSOLVE_READMOSTLY Procedure	This procedure dissolves an object's read-mostly setting in an Oracle RAC environment.
GRAB_AFFINITY Procedure	This procedure sets the object affinity to a node in an Oracle RAC environment.
GRAB_READMOSTLY Procedure	This procedure grabs an object's read-mostly setting in an Oracle RAC environment.
LIST_READMOSTLY Procedure	This procedure lists objects that have the read-mostly property set in an Oracle RAC environment.
OBJECT_DOWNCONVERT Procedure	This procedure down-converts an object's exclusive locks on an instance to shared mode for all instances in an Oracle RAC environment.
TRUE_CACHE_KEEP Procedure	When you call this procedure on a True Cache instance, it assigns the object to the `KEEP` buffer pool on that instance.
TRUE_CACHE_UNKEEP Procedure	When an object on a True Cache instance no longer needs to be in the `KEEP` buffer pool, use this procedure to remove the object's `KEEP` assignment.

DISSOLVE_AFFINITY Procedure

`DISSOLVE_AFFINITY` dissolves the object affinity from a node in an Oracle RAC environment.

Syntax

```
DBMS_CACHEUTIL.DISSOLVE_AFFINITY(
   schema              IN VARCHAR2,
   obj                 IN VARCHAR2,
   partition           IN VARCHAR2 := NULL,
   dissolve_index      IN BOOLEAN := TRUE,
   active_drm          IN BOOLEAN := FALSE);
```

Parameters

Table 46-2 DISSOLVE_AFFINITY Procedure Parameters

Parameter	Description
`schema`	The name of the schema for the object.
`obj`	The name of the object.
`partition`	• If the object is not partitioned, then this is `NULL`. • If the object is partitioned, use the name of the partition segment. • If it's a composited partitioned object, use the name of the subpartition segment.
`dissolve_index`	To dissolve affinity for dependent indexes, use `TRUE`.
`active_drm`	For internal use only. Do not use.

DISSOLVE_READMOSTLY Procedure

`DISSOLVE_READMOSTLY` dissolves an object's read-mostly setting in an Oracle RAC environment.

Syntax

```
DBMS_CACHEUTIL.DISSOLVE_READMOSTLY(
   schema              IN VARCHAR2,
   obj                 IN VARCHAR2,
   partition           IN VARCHAR2 := NULL,
   dissolve_index      IN BOOLEAN := TRUE);
```

Parameters

Table 46-3 DISSOLVE_READMOSTLY Procedure Parameters

Parameter	Description
`schema`	The name of the schema for the object.
`obj`	The name of the object.

Table 46-3 (Cont.) DISSOLVE_READMOSTLY Procedure Parameters

Parameter	Description
partition	• If the object is not partitioned, then this is NULL. • If the object is partitioned, use the name of the partition segment. • If it's a composited partitioned object, use the name of the subpartition segment.
dissolve_index	To dissolve read-mostly for dependent indexes, use TRUE.

GRAB_AFFINITY Procedure

GRAB_AFFINITY sets the object affinity to a node in an Oracle RAC environment.

This is useful when you know that, for a particular workload, an object will be heavily accessed by one particular Oracle RAC node. Grabbing an object affinity to that node reduces locking overhead and could improve performance.

> **Note:**
>
> The affinity that this procedure grabs can still change by the automatic object affinity policy if the policy decides that it's not optimal to keep the affinity. The CURRENT_MASTER column in the V$GCSPFMASTER_INFO view indicates the current affinity of the particular object segment.

> **Note:**
>
> Object affinity locking optimization works only for the primary RAC database, and it doesn't work for Oracle Active Data Guard.

Syntax

```
DBMS_CACHEUTIL.GRAB_AFFINITY(
   schema              IN VARCHAR2,
   obj                 IN VARCHAR2,
   partition           IN VARCHAR2 := NULL,
   grab_index          IN BOOLEAN := TRUE,
   active_drm          IN BOOLEAN := FALSE);
```

Parameters

Table 46-4 GRAB_AFFINITY Procedure Parameters

Parameter	Description
schema	The name of the schema for the object.
obj	The name of the object.

Table 46-4 (Cont.) GRAB_AFFINITY Procedure Parameters

Parameter	Description
partition	• If the object is not partitioned, then this is NULL. • If the object is partitioned, use the name of the partition segment. • If it's a composited partitioned object, use the name of the subpartition segment.
grab_index	To grab affinity for dependent indexes, use TRUE.
active_drm	For internal use only. Do not use.

GRAB_READMOSTLY Procedure

GRAB_READMOSTLY sets an object to read-mostly in an Oracle RAC environment.

This is useful when you know that, for a particular workload, accesses to an object will mostly be READ instead of INSERT or UPDATE. Grabbing read-mostly for the object would reduce locking overhead and could improve performance.

> **Note:**
>
> The read-mostly that this procedure grabs can still change by the automatic read-mostly policy if the policy decides that it's not optimal to keep the read-mostly set. The read-mostly property is persistent across instance lifetimes. You can use the LIST_READMOSTLY procedure included in this package to list the current set of object segments with read-mostly set.

> **Note:**
>
> Read-mostly locking optimization works only for the primary RAC database, and it doesn't work for Oracle Active Data Guard.

Syntax

```
DBMS_CACHEUTIL.GRAB_READMOSTLY(
   schema              IN VARCHAR2,
   obj                 IN VARCHAR2,
   partition           IN VARCHAR2 := NULL,
   grab_index          IN BOOLEAN := TRUE);
```

Parameters

Table 46-5 GRAB_READMOSTLY Procedure Parameters

Parameter	Description
schema	The name of the schema for the object.
obj	The name of the object.

Table 46-5 (Cont.) GRAB_READMOSTLY Procedure Parameters

Parameter	Description
partition	• If the object is not partitioned, then this is NULL. • If the object is partitioned, use the name of the partition segment. • If it's a composited partitioned object, use the name of the subpartition segment.
grab_index	To grab read-mostly for dependent indexes, use TRUE.

LIST_READMOSTLY Procedure

This procedure lists objects that have the read-mostly property set in an Oracle RAC environment.

Syntax

```
DBMS_CACHEUTIL.LIST_READMOSTLY;
```

OBJECT_DOWNCONVERT Procedure

OBJECT_DOWNCONVERT down-converts an object's exclusive locks on an instance to shared mode for all instances in an Oracle RAC environment.

This is useful when a particular object is loaded through the buffer cache by an Oracle RAC node and this node holds many exclusive locks for this object, which has caused extra locking overhead. Down-converting locks for this object to shared locks on the RAC node could reduce the locking overhead and improve performance.

Syntax

```
DBMS_CACHEUTIL.OBJECT_DOWNCONVERT(
   schema                IN VARCHAR2,
   obj                   IN VARCHAR2,
   partition             IN VARCHAR2 := NULL,
   downconvert_index     IN BOOLEAN := TRUE);
```

Parameters

Table 46-6 OBJECT_DOWNCONVERT Procedure Parameters

Parameter	Description
schema	The name of the schema for the object.
obj	The name of the object.
partition	• If the object is not partitioned, then this is NULL. • If the object is partitioned, use the name of the partition segment. • If it's a composited partitioned object, use the name of the subpartition segment.
downconvert_index	To down-convert dependent indexes, use TRUE.

TRUE_CACHE_KEEP Procedure

When you call this procedure on True Cache, it assigns the object to the `KEEP` buffer pool on that cache.

To use this procedure, the `DB_KEEP_CACHE_SIZE` initialization parameter must be configured on True Cache.

Syntax

```
DBMS_CACHEUTIL.TRUE_CACHE_KEEP(
   schema                 IN VARCHAR2,
   obj                    IN VARCHAR2,
   partition              IN VARCHAR2 := NULL);
```

Parameters

Table 46-7 TRUE_CACHE_KEEP Procedure Parameters

Parameter	Description
`schema`	The name of the schema for the object.
`obj`	The name of the object.
`partition`	• If the object is not partitioned, then this is `NULL`. • If the object is partitioned, use the name of the partition segment. • If it's a composited partitioned object, use the name of the subpartition segment.

Related Topics

- Oracle True Cache User's Guide

TRUE_CACHE_UNKEEP Procedure

When an object on True Cache no longer needs to be in the `KEEP` buffer pool, use this procedure to remove the object's `KEEP` assignment.

Syntax

```
DBMS_CACHEUTIL.TRUE_CACHE_UNKEEP(
   schema                 IN VARCHAR2,
   obj                    IN VARCHAR2,
   partition              IN VARCHAR2 := NULL);
```

Parameters

Table 46-8 TRUE_CACHE_UNKEEP Procedure Parameters

Parameter	Description
`schema`	The name of the schema for the object.
`obj`	The name of the object.

Table 46-8 (Cont.) TRUE_CACHE_UNKEEP Procedure Parameters

Parameter	Description
`partition`	• If the object is not partitioned, then this is `NULL`. • If the object is partitioned, use the name of the partition segment. • If it's a composited partitioned object, use the name of the subpartition segment.

Related Topics

- Oracle True Cache User's Guide