16

Using Persistent Memory Database

Mapping the database directly into persistent memory (PMEM) provides significant performance enhancements.

16.1 About Persistent Memory Database

The Persistent Memory Database feature includes directly mapped buffer cache and Persistent Memory Filestore (PMEM Filestore).

16.1.1 What Is Persistent Memory Database?

The Persistent Memory Database feature enables you to place database files in non-volatile memory. This feature supports a single-instance Oracle Database instance on PMEM Filestore.

16.1.2 What Is Oracle Persistent Memory Filestore?

PMEM Filestore is a pointer-switching PMEM file system that supports atomic updates of Oracle Database data blocks. PMEM Filestore is the underlying file store used for a Persistent Memory database. PMEM Filestore provides the external interface for mapping and accessing an Oracle database directly in persistent memory.

Managing an Oracle database on PMEM Filestore is similar to managing an Oracle database on a native file system. PMEM Filestore implements the Filesystem in Userspace (FUSE) protocol, enabling Oracle DBAs to perform normal file-level maintenance. FUSE allows non-privileged (non-root) users, such as the Oracle Database software owner, to create and manage filesystems as well as the directories and files contained within them.

A typical file system uses raw storage as its backing store, while PMEM Filestore gets storage from a native operating system file in a PMEM DAX file system. This file is called the backing file and is visible as a file in the operating system. PMEM Filestore subdivides the storage within the backing file and presents it as a local file system.

After a PMEM Filestore is created and mounted, you will see a local file system under the user-specified mount point. This local file system supports directories and common operating system commands such as `ls` and `cp`. This local file system is the PMEM Filestore and can be used to store Oracle Database files. Note that the PMEM Filestore is only visible when the Oracle Database instance is started.

You can use PMEM Filestore for database datafiles and control files. For performance reasons, Oracle recommends that you store redo log files as independent files in a DAX-aware filesystem such as EXT4/XFS. Administrative files such as trace files and audit files cannot be stored in PMEM Filestore. The server parameter file (SPFILE) cannot be stored in PMEM Filestore because PMEM Filestore configuration parameters can be specified in the SPFILE.

16.1.3 What Is Directly Mapped Buffer Cache?

Directly mapped buffer cache is a mechanism in Oracle Database to directly read data on persistent memory, bypassing the traditional DRAM buffer cache. This mechanism also tracks data access and automatically brings frequently read data, and data for updating, from PMEM to DRAM buffer cache for faster access. The directly mapped buffer cache mechanism is automatically invoked when a datafile is placed in a PMEM filestore.

16.1.4 Benefits of Using Persistent Memory Database

With Oracle Persistent Memory Database, database files can be placed in persistent memory which enables Oracle to take advantage of performance enhancements inherent to this technology. Consider the following benefits of using Persistent Memory Database:

- The PMEM Filestore provides atomic writes to full Oracle database blocks. This eliminates the need for media recovery due to partially written blocks after a power outage.
- Persistent Memory Database performs I/O to PMEM storage via memory copy. This is much faster than performing I/O via traditional operating system calls.
- Database queries save the traditional read from storage and memory copy into buffer cache because the Oracle Database server accesses data directly from persistent memory.

16.2 Setting Initialization Parameters for Persistent Memory Database

You can set the `PMEM_FILESTORE` initialization parameter to specify a PMEM Filestore that the Oracle Database instance will mount automatically when it is started.

- Persistent Memory Database Initialization Parameters

16.2.1 Persistent Memory Database Initialization Parameters

The `PMEM_FILESTORE` initialization parameter specifies a PMEM Filestore that the Oracle Database instance will automatically mount when it is started. The parameter is set to an ordered pair of strings. The first string in the parameter value list is the directory where PMEM Filestore is mounted. The second string is the backing file.

A PMEM Filestore backing file is visible as a file in the operating system file system hierarchy. While a typical file system uses raw storage as its backing store, PMEM Filestore gets storage from a native operating system file in a PMEM DAX file system. On Linux, the PMEM Filestore backing file should be in an XFS or ext4 file system mounted using the `-o dax` option.

16.3 Creating a PMEM Filestore for an Oracle Database

To use the Persistent Memory Database feature, you create a PMEM filestore for Oracle Database files.

- Creating a PMEM Filestore Before Creating the Database
- Creating a Database on PMEM Storage Using Oracle DBCA
  Starting with Oracle Database 23ai, you can use Oracle Database Configuration Assistant (Oracle DBCA), in either interactive or silent mode, to select persistent memory (PMEM) as your storage option when creating a single-instance database.
- Creating an Oracle Database in the PMEM Filestore
- Migrating an Oracle Database to a PMEM Filestore

16.3.1 Creating a PMEM Filestore Before Creating the Database

Perform the following steps to create a PMEM filestore for an Oracle database:

1. Start the Oracle Database instance in `NOMOUNT` mode.
2. Execute the `CREATE PMEM FILESTORE` command to create the PMEM filestore and provide:
   - A mount point for the file store. The final subdirectory name must be the same as the PMEM filestore name.
   - A backing file from a native XFS or ext4 file system mounted in DAX mode.The backing file is used by the PMEM filestore to keep all the files created in the filestore.
   - A block size, which will typically be the same as the default block size of the database datafiles.
     As an example:

     ```
     CREATE PMEM FILESTORE db1_pmemfs
     MOUNTPOINT '/u1/db/db1_pmemfs'
     BACKINGFILE '/u1/db_storage/db1'
     SIZE 2T
     BLOCKSIZE 8K
     AUTOEXTEND ON NEXT 10G
     MAXSIZE 3T;
     ```

     The PMEM filestore is automatically mounted after it is created. The PMEM filestore will appear under the specified mount point as if it is a native file system.
3. Configure the PMEM filestore to be mounted during instance startup.
   If you used an SPFILE to start the database instance, the server adds the `PMEM_FILESTORE` initialization parameter to the SPFILE. This parameter causes the PMEM filestore to be automatically mounted when the database instance is started. The SPFILE will contain this entry following the execution of the `CREATE PMEM FILESTORE` command: `PMEM_FILESTORE=(‘/u1/db/db1_pmemfs’, ‘/u1/db_storage/db1’)`. The first string in the parameter value list is the directory where the PMEM filestore is mounted. The second string is the backing file.

If you did not use an SPFILE to start the database instance, you must manually add the `PMEM_FILESTORE` initialization parameter so that the PMEM filestore is mounted during instance startup. Or you must manually mount the PMEM filestore using the `ALTER PMEM FILESTORE … MOUNT` command.

16.3.2 Creating a Database on PMEM Storage Using Oracle DBCA

Starting with Oracle Database 23ai, you can use Oracle Database Configuration Assistant (Oracle DBCA), in either interactive or silent mode, to select persistent memory (PMEM) as your storage option when creating a single-instance database.

Perform the following steps to create a PMEM filestore for an Oracle database using DBCA:

1. Start Oracle Database Configuration Assistant (Oracle DBCA).

   ```
   $ cd $ORACLE_HOME/bin
   $ ./dbca
   ```

2. In the **Select Database Operation** screen, select **Create a database** and click **Next**.
3. In the **Select Database Creation Mode** screen, select **Advanced configuration** and click **Next**.
4. In the **Select Database Deployment type** screen, select **Oracle single instance database**, as the Database type. Select an appropriate template for your database. Click **Next**.
5. In the **Specify Database Identification Details** page, for the **Global database name**, enter the database name in the form `database_name.domain_name`. In the `SID` field, enter the system identifier. The `SID` defaults to the database name and uniquely identifies the instance that runs the database.
   You can choose to create either an empty container database (CDB) or a CDB with one or more pluggable databases (PDB). Enter the number of PDBs to create in the **Number of PDBs** field. In the **PDB Name** field, specify a name to use for the PDB or PDBs to be created. When you create multiple PDBs, the PDB name you specify is used as a prefix for the PDBs to be created. Click **Next**.
6. In the **Select Database Storage Option** screen, select **Use following for the database storage attributes**.
   Specify the Database files storage type as **PMEM File System**. Browse to specify the location of your PMEM storage in the **Database files location** field.

   Specify the **PMEM File System**. The PMEM File System can use either a PMEM filestore, a DAX-enabled file system, or the Oracle Memory Speed file system. Specify the PMEM File System Size.

   Select **Use Oracle-Managed Files (OMF)**, to directly manage operating system files comprising an Oracle Database.
7. Respond to the configuration screens and prompts as needed to complete the database creation process. Configuration screens vary depending on the configuration option that you select.

> **Note:**
>
> Click **Help** if you have any questions about the information you are asked to submit during database creation.

16.3.3 Creating an Oracle Database in the PMEM Filestore

After creating the PMEM filestore, you can use it as if it is a native file system to create an Oracle Database. You can create the database under the mount point you specified when you created the PMEM filestore. See *Oracle Multitenant Administrator's Guide* for detailed information.

16.3.4 Migrating an Oracle Database to a PMEM Filestore

After creating the PMEM filestore, perform the following steps to migrate an existing Oracle Database or select tablespaces to the PMEM filestore:

1. Copy the database files by using the RMAN `RESTORE` command or operating system commands.
2. Optionally, change the block size of the redo log files by creating new online redo log files and then dropping the existing redo log files.

You can also use the `ALTER DATABASE` command with the `MOVE DATAFILE` clause to move an online datafile to the PMEM filestore.

16.4 Managing a PMEM Filestore

You can view information about a PMEM filestore and perform various operations on the PMEM filestore, including mounting and dismounting, changing the attributes, and dropping the filestore.

- Viewing Information About a PMEM Filestore
  `V$PMEM_FILESTORE` provides information about the PMEM filestore.
- Mounting a PMEM Filestore
  If you did not set the `PMEM_FILESTORE` initialization parameter to automatically mount the PMEM filestore when the database instance is started, you must mount the PMEM filestore manually.
- Dismounting a PMEM Filestore
  If you want to alter the attributes of the PMEM filestore, you must first dismount the filestore.
- Changing the Attributes of a PMEM Filestore
  You can change the attributes of a PMEM filestore, including the mount point, the backing file, and the size.
- Dropping a PMEM Filestore
  You can drop a PMEM filestore whether it is empty or not.

16.4.1 Viewing Information About a PMEM Filestore

`V$PMEM_FILESTORE` provides information about the PMEM filestore.

You can query `V$PMEM_FILESTORE` to view information about the PMEM filestore including:

- Directory path for the mount point of the PMEM filestore
- File path for the backing file of the PMEM filestore
- Block size of the PMEM filestore (in bytes)

- Current size of the PMEM filestore (in bytes)
- Whether it is autoextensible and details about the autoextensible configuration
- Space usage (free space and used space)

16.4.2 Mounting a PMEM Filestore

If you did not set the `PMEM_FILESTORE` initialization parameter to automatically mount the PMEM filestore when the database instance is started, you must mount the PMEM filestore manually.

The following statement is used to mount the PMEM filestore:

- `ALTER PMEM FILESTORE` *`filestore_name`* `MOUNT`

If the initialization parameter file does not include the `PMEM_FILESTORE` parameter, you must include the `MOUNTPOINT` and `BACKINGFILE` clauses, along with the `FORCE` keyword when you execute this command.

You can also include the `MOUNTPOINT` and `BACKINGFILE` clauses with the `FORCE` keyword to specify different file paths than were specified in the intialization parameter file. If you used a server parameter file (SPFILE) to start the database instance, it will be updated with the file paths specified in the `ALTER PMEM FILESTORE MOUNT` command.

You must be connected to the root container (`CDB$ROOT`) as a user with the `SYSDBA` privilege to execute this command.

16.4.3 Dismounting a PMEM Filestore

If you want to alter the attributes of the PMEM filestore, you must first dismount the filestore.

The following statement is used to dismount the PMEM filestore:

- `ALTER PMEM FILESTORE` *`filestore_name`* `DISMOUNT`

You must be connected to the root container (`CDB$ROOT`) as a user with the `SYSDBA` privilege to execute this command.

16.4.4 Changing the Attributes of a PMEM Filestore

You can change the attributes of a PMEM filestore, including the mount point, the backing file, and the size.

The following statement is used to change the attributes of the PMEM filestore:

- `ALTER PMEM FILESTORE`

If you want to change the mount point or backing file, you must first dismount the filestore. See Dismounting a PMEM Filestore.

You cannot change the block size of the filestore.

You must be connected to the root container (`CDB$ROOT`) as a user with the `SYSDBA` privilege to execute this command.

16.4.5 Dropping a PMEM Filestore

You can drop a PMEM filestore whether it is empty or not.

The following statement is used to drop the PMEM filestore:

- `DROP PMEM FILESTORE`

Specify `INCLUDING CONTENTS` to remove all of the files in the PMEM filestore. To drop the filestore immediately, use the `FORCE` keyword with `INCLUDING CONTENTS`. Specify `EXCLUDING CONTENTS` so that the PMEM filestore will be dropped only when it is empty.

If the database instance was started with an SPFILE, the SPFILE will be updated when `DROP PMEM FILESTORE` is executed.