Database Memory Allocation

This chapter describes memory allocation in Oracle Database and the various methods for managing memory.

This chapter contains the following topics:

- About Database Memory Caches and Other Memory Structures
- Database Memory Management Methods
- Using Automatic Memory Management
- Monitoring Memory Management

About Database Memory Caches and Other Memory Structures

Oracle Database stores information in memory caches and on disk. Memory access is much faster than disk access. Disk access (physical I/O) takes a significant amount of time, compared to memory access, typically in the order of 10 milliseconds. Physical I/O also increases the CPU resources required due to the path length in device drivers and operating system event schedulers. For this reason, it is more efficient for data requests of frequently accessed objects to be perform by memory, rather than also requiring disk access. Proper sizing and effective use of Oracle Database memory caches greatly improves database performance.

The main Oracle Database memory caches that affect performance include:

Database buffer cache

The database buffer cache stores data blocks read from disk.

Redo log buffer

The redo log buffer stores redo entries of changes made to data blocks in the buffer cache.

Shared pool

The shared pool caches many different types of data and is mainly comprised of the following components:

- Library cache
- Data dictionary cache
- Server result cache
- Large pool

The large pool provides large memory allocations for the following Oracle Database features:

- Shared server architecture
- Parallel query
- Recovery Manager (RMAN)
- Java pool



The Java pool stores session-specific Java code and Java Virtual Machine (JVM) data.

Streams pool

The Streams pool provides memory for Oracle Advanced Queuing (AQ) and replication processes.

Process-private memory

Process-private memory includes memory used for operations such as sorting and hash joins.

In-Memory Column Store (IM column store)

The IM column store is an optional, static SGA pool that stores copies of tables and partitions. In the IM column store, data is stored in a special columnar format, which improves performance of operations such as scans, joins, and aggregations.



The IM column store does not replace the buffer cache, but acts as a supplement so that both memory areas can store the same data in different formats.

 Unified memory allows better usage of memory. Use memory_size rather than sga_target and pga aggregate target/limit.

See Also:

Oracle Database Concepts for information about the Oracle Database memory architecture

Database Memory Management Methods

The goal of memory management is to reduce the physical I/O overhead as much as possible, either by making it more likely that the required data is in memory, or by making the process of retrieving the required data more efficient. To achieve this goal, proper sizing and effective use of Oracle Database memory caches is essential.

Figure 12-1shows a decision tree for memory management options. The following sections explain the options in detail.

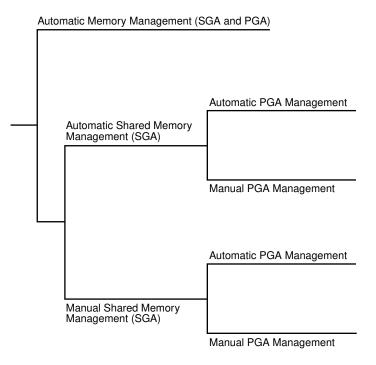


Figure 12-1 Memory Management Methods

- Automatic Memory Management
- Automatic Shared Memory Management
- Manual Shared Memory Management
- Automatic PGA Memory Management
- Manual PGA Memory Management
- Summary of Memory Management Methods

Automatic Memory Management

In **automatic memory management**, Oracle Database manages the shared global area (SGA) and program global area (instance PGA) memory completely. This method is the simplest and is strongly recommended by Oracle.

Before setting any memory pool sizes manually, strongly consider using automatic memory management.

The only user-specified controls are the target memory size initialization parameter (MEMORY_TARGET) and optional maximum memory size initialization parameter (MEMORY_MAX_TARGET). Oracle Database tunes to the target memory size, redistributing memory as needed between the SGA and the instance PGA.

The SGA can contain the In-Memory Column Store (IM column store) or memoptimize pool, both of which are optional. No matter which memory management method you use, size the IM column store with the INMEMORY_SIZE initialization parameter, and the memoptimize pool with the MEMOPTIMIZE_POOL_SIZE initialization parameter. The IM column store size are memoptimize pool are included in the memory target, but are not managed by the automatic resize algorithm. For example, if you set MEMORY_TARGET to 5 GB and INMEMORY_SIZE to 1 GB, then the overall memory target is 5 GB (not 6 GB), and the INMEMORY_SIZE is always 1 GB.

The following graphic shows a database that sometimes processes jobs submitted by online users and sometimes batch jobs. Using automatic memory management, the database automatically adjusts the size of the large pool and database buffer cache depending on which type of jobs are running. Note that the graphic does not depict the IM column store or memoptimize area.

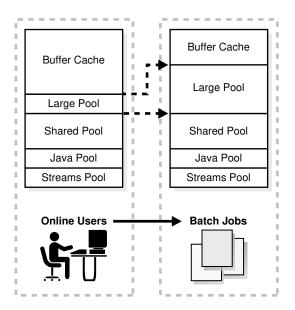


Figure 12-2 Automatic Memory Management

If you create a database with DBCA and choose the basic installation option, then Oracle Database enables automatic memory management by default.

See Also:

- Using Automatic Memory Management
- Oracle Database Administrator's Guide to learn about automatic memory management

Automatic Shared Memory Management

Oracle Database uses automatic shared memory management of the SGA memory when automatic memory management is disabled.

In this mode, Oracle Database automatically distributes memory to individual SGA components based on a target size that you set for the total SGA memory.

The database tunes the total SGA to the target size and dynamically tunes the sizes of SGA components. If you are using a server parameter file, then Oracle Database remembers the sizes of the automatically tuned components across database instance shutdowns.

For information about using automatic shared memory management, see "Using Automatic Shared Memory Management".

Manual Shared Memory Management

If both automatic memory management and automatic shared memory management are disabled, then you must manage SGA memory manually by sizing the individual memory pools in the SGA. Although this mode enables you to exercise complete control over how SGA memory is distributed, it requires the most effort because the SGA components must be manually tuned on an ongoing basis.



When automatic memory management is disabled, then in some cases the database can automatically adjust the relative sizes of the shared pool and buffer cache, based on user workload.

For information about using manual shared memory management, see "Sizing the SGA Components Manually".

Automatic PGA Memory Management

When automatic memory management (MEMORY_TARGET) is disabled and PGA_AGGREGATE_TARGET is set to a nonzero value, then Oracle Database uses automatic PGA memory management. In this mode, PGA_AGGREGATE_TARGET specifies a "soft" target size for the instance PGA. The target is soft because it only applies to specific types of memory allocations that can choose to use temporary space rather than the PGA. The database tunes the size of the instance PGA to this target and dynamically tunes the sizes of individual PGAs. If you do not explicitly set a target size, then the database automatically configures a reasonable default.

The PGA_AGGREGATE_LIMIT initialization parameter dynamically sets an instance-wide hard limit for PGA memory. Because the parameter responds to changing memory conditions, you do not need to set the parameter value explicitly. By default, PGA_AGGREGATE_LIMIT is set to the greater of the following:

- 2 GB
- 200% of PGA AGGREGATE TARGET initialization parameter setting
- (Value of PROCESSES initialization parameter setting) * 3 MB

A background process periodically compares the PGA size to the limit set by PGA_AGGREGATE_LIMIT. If the limit is reached or exceeded, then this process terminates calls for the sessions using the most untunable PGA memory. If these sessions still do not release enough memory, then they are also terminated.

For information about automatic PGA memory management, see Tuning the Program Global Area .

Manual PGA Memory Management

When automatic memory management is disabled and PGA_AGGREGATE_TARGET is set to 0 (automatic PGA memory management is disabled), the database defaults to manual PGA management. You must manage PGA memory manually by adjusting the portion of PGA memory dedicated to each work area.

This method can be very difficult because the workload is always changing and is not recommended by Oracle. Although manual PGA memory management is supported by Oracle Database, Oracle strongly recommends using automatic memory management or automatic PGA memory management instead.

Summary of Memory Management Methods

Memory management is either automatic or manual.

If you do not enable automatic memory management, then you must separately configure one memory management method for the SGA and one for the PGA.



When automatic memory management is disabled for the database instance as a whole, Oracle Database enables automatic PGA memory management by default.

The following table includes the <code>INMEMORY_SIZE</code> initialization parameter and the <code>MEMOPTIMIZE_POOL_SIZE</code> initialization parameter. Both control memory areas that are optional.

Table 12-1 Memory Management Methods

Instance	SGA	PGA	Description	Initialization Parameters
Auto	n/a	n/a	The database tunes the size of the instance based on a single instance target size.	You set: Total memory target size for the database instance (MEMORY_TARGET) Optional maximum memory size for the database instance (MEMORY_MAX_TARGET) Optional size for the IM column store (INMEMORY_SIZE) in the SGA
n/a	Auto	Auto	The database automatically tunes the SGA based on an SGA target. The database automatically tunes the PGA based on a PGA target.	You set: SGA target size (SGA_TARGET) Optional SGA maximum size (SGA_MAX_SIZE) Optional size for the IM column store (INMEMORY_SIZE) in the SGA Optional size for the memoptimize pool in the SGA (MEMOPTIMIZE_POOL_SIZE) PGA aggregate target size (PGA_AGGREGATE_TARGET)¹ The database automatically configures the PGA_AGGREGATE_LIMIT initialization parameter. You may set this parameter manually.



Table 12-1 (Cont.) Memory Management Methods

Instance	SGA	PGA	Description	Initialization Parameters
n/a	Auto	Manual	The database automatically tunes the SGA based on an SGA target. You control the PGA manually, setting the maximum work area size for each type of SQL operator.	You set: SGA target size (SGA_TARGET) Optional SGA maximum size (SGA_MAX_SIZE) Optional size for the IM column store in the SGA (INMEMORY_SIZE) Optional size for the memoptimize pool in the SGA (MEMOPTIMIZE_POOL_SIZE) PGA work area parameters such as SORT_AREA_SIZE, HASH_AREA_SIZE, and BITMAP MERGE AREA SIZE
n/a	Manual	Auto	You control the SGA manually by setting individual component sizes. The database automatically tunes the PGA based on a PGA target.	You set: Shared pool size (SHARED_POOL_SIZE) Buffer cache size (DB_CACHE_SIZE) Large pool size (LARGE_POOL_SIZE) Java pool size (JAVA_POOL_SIZE) Optional size for the IM column store (INMEMORY_SIZE) in the SGA Optional size for the memoptimize pool in the SGA (MEMOPTIMIZE_POOL_SIZE) PGA aggregate target size (PGA_AGGREGATE_TARGET) ² The database automatically configures the PGA_AGGREGATE_LIMIT initialization parameter. You may set this parameter manually.
n/a	Manual	Manual	You must manually configure SGA component sizes. You control the PGA manually, setting the maximum work area size for each type of SQL operator.	You must manually configure SGA component sizes. You set: Shared pool size (SHARED_POOL_SIZE) Buffer cache size (DB_CACHE_SIZE) Large pool size (LARGE_POOL_SIZE) Java pool size (JAVA_POOL_SIZE) Optional size for the IM column store (INMEMORY_SIZE) in the SGA Optional size for the memoptimize pool in the SGA (MEMOPTIMIZE_POOL_SIZE) PGA work area parameters such as SORT_AREA_SIZE, HASH_AREA_SIZE, and BITMAP_MERGE_AREA_SIZE

¹ The database automatically configures the PGA_AGGREGATE_LIMIT initialization parameter. You also choose to set this parameter manually.

See Also:

Oracle Database Administrator's Guide because automatic memory management is not available on all platforms

² The database automatically configures the PGA_AGGREGATE_LIMIT initialization parameter. You also choose to set this parameter manually.

Using Automatic Memory Management

To use automatic memory management, set the following initialization parameters:

• MEMORY TARGET

The MEMORY_TARGET initialization parameter specifies the target memory size. The database tunes to the value specified for this parameter, redistributing memory as needed between the SGA and the instance PGA. This parameter is dynamic, so its value can be changed at any time without restarting the database.

• MEMORY MAX TARGET

The MEMORY_MAX_TARGET initialization parameter specifies the maximum memory size. The value specified for this parameter serves as the limit to which the MEMORY_TARGET initialization parameter can be set. This parameter is static, so its value cannot be changed after instance startup.

If you need tuning advice for the MEMORY_TARGET parameter, then use the V\$MEMORY TARGET ADVICE view.



Oracle Database Administrator's Guide for information about using automatic memory management

Monitoring Memory Management

Table 12-2 lists the views that provide information about memory resize operations.

Table 12-2 Memory Management Views

View	Description
V\$MEMORY_CURRENT_RESIZE_OPS	Displays information about memory resize operations (both automatic and manual) that are currently in progress.
V\$MEMORY_DYNAMIC_COMPONENTS	Displays information about the current sizes of all dynamically-tuned memory components, including the total sizes of the SGA and instance PGA.
V\$MEMORY_RESIZE_OPS	Displays information about the last 800 completed memory resize operations (both automatic and manual). This does not include operations that are currently in progress.
V\$MEMORY_TARGET_ADVICE	Displays tuning advice for the MEMORY_TARGET initialization parameter.

See Also

Oracle Database Reference for more information about these views