Altibase White Paper

User Memory Tablespaces Overview

ALTIBASE® HDB™ Hybrid RDBMS

Publication Date: May 2011
Author: Cy Erbay, Principal Technologist, Altibase, Inc.

Table of Contents

Introduction

Your organization's mission-critical applications need extremely fast transaction processing speeds to handle big data. Vast amounts of digital information makes it possible for you to do things that previously could not be done, as long as you have access to the data in real time. Managed well, big data provides insights into business trends, avoiding supply chain shortages, uncovering criminal activities, discovering medical breakthroughs, and many other emerging sources of economic value. Accessing and analyzing this data in real-time is no longer optional.

In today's fast paced business environment, operational efficiencies demand speed. Your customers, partners, suppliers and employees expect instantaneous response. Continuous access to real-time data is essential to gaining your competitive advantage. The future of your company, your department and your career depends upon fast, real-time access and analysis of high volumes of data in mission critical environments.
To meet the demands of today's modern IT environments, Altibase Corporation has taken an innovative approach by combining In-Memory (IMDB) and Disk Resident (DRBM) database concepts into a single, full-featured RDBMS, ALTIBASE HDB.

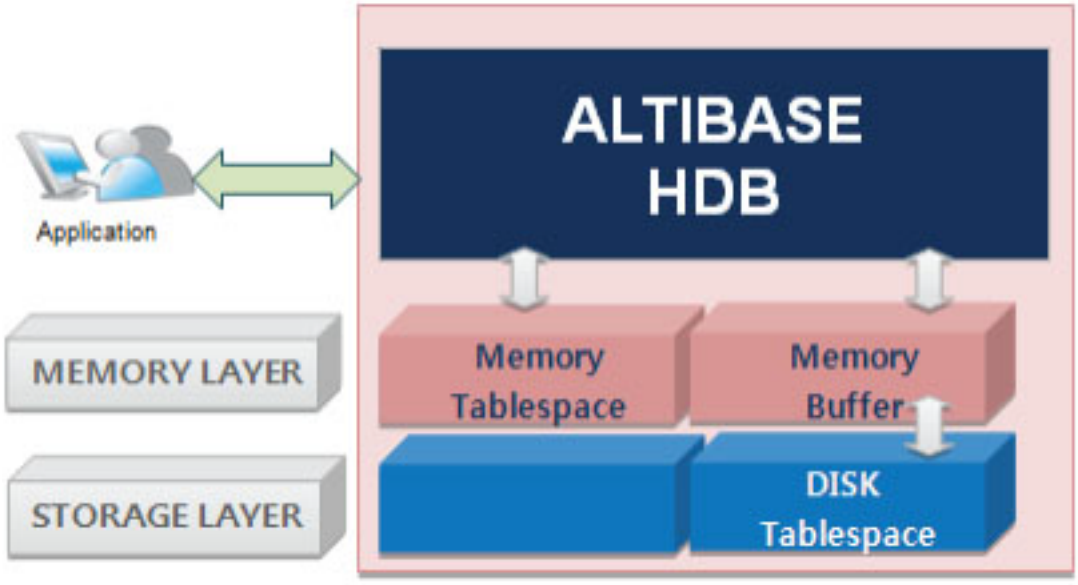

Figure 1 – ALTIBASE HDB = IMDB + DRDB

ALTIBASE HDB is a hybrid relational DBMS that delivers extreme speed while supporting large data sets. ALTIBASE HDB reliably supports real-time applications and allows information managers to pick and choose between in-memory and on-disk data storage models.

With the in-memory database component of ALTIBASE HDB, the entire data resides in memory providing extreme performance, predictable response times, high throughput, and low latency without any disk I/O overhead and with no compromise from ACID (Atomicity, Consistency, Isolation, Durability) properties that are expected from an enterprise level database solution. ALTIBASE HDB In-Memory database is persistent and recoverable.

At the core of ALTIBASE HDB In-Memory database is the user memory tablespace which is used for managing mission-critical, frequently accessed hot data. The user memory tablespace allows database administrators to manage massive amounts of memory data efficiently and safely.

This white paper is provided for those that desire a deeper understanding of how ALTIBASE HDB user memory tablespaces are designed and function.

Types of ALTIBASE HDB Tablespaces

ALTIBASE HDB tablespace is a logical storage unit for storing tables, indexes and other database objects.

There are two groups of tablespaces in ALTIBASE HDB, system and user-defined.
When a database is first created, Altibase HDB automatically creates a tablespace group called system that contains general information about the structure and the contents of the database.

The second group of the tablespaces is called user (user-defined) tablespaces, and they are created by database users.

ALTIBASE HDB supports 3 types of user tablespaces:
- Disk tablespaces
- Memory tablespaces
- Volatile tablespaces

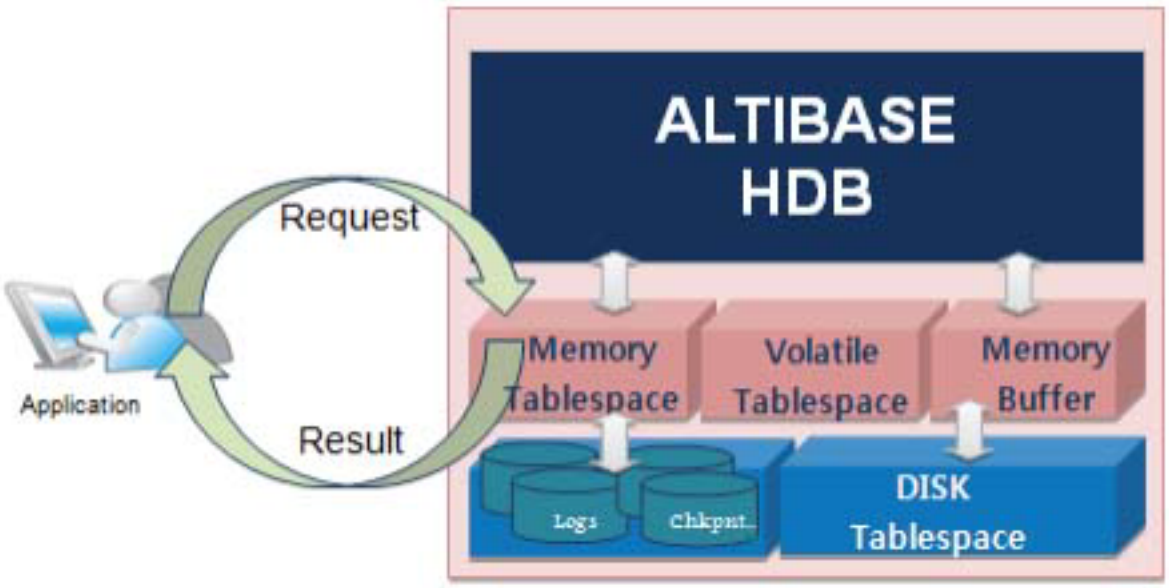

Figure 2 – ALTIBASE HDB Tablespaces

In case of user disk tablespaces, the data resides on the disk file system, and if desired, it can be loaded in the memory buffer pool.

On the other hand, with both user memory and user volatile tablespaces, the entire data resides in main memory.

User memory tablespaces save all backup images and changes to the data on the disk to guarantee data integrity. In case of a database restart, all the data is loaded back into memory.

The user volatile tablespace is similar to the user memory tablespace, but it does not guarantee data integrity. In case of a database restart, unlike the memory tablespace, all data in the volatile memory tablespace is lost.

When determining whether to create a memory, disk or volatile tablespace, the user should consider the characteristics of the objects to be stored in the tablespace, such as their size and the frequency at which they are accessed.

While user disk tablespaces are an appropriate choice for large volumes of data, such as historical cold data, memory tablespaces are more suitable for small volumes of data that is accessed frequently. Volatile tablespaces enable users for highest performance data access but compromise from data integrity.

This paper will focus on user-defined tablespaces, specifically user memory tablespaces

Memory Tablespace Functionality

User memory tablespaces allow independence from disk I/O operations while still supporting an identical interface to the disk tablespaces. However, user memory tablespaces allow ultra-high performance for accessing data in the main memory with minimal overhead, thus they are a perfect match for applications requiring near-real time access to time-critical data.

The ALTIBASE HDB users can perform all common database operations on memory tablespaces such as create, delete, change, backup and restore. Users can also control the states of memory tablespaces, such as online or offline states.

Memory Tablespace Structure

The ALTIBASE HDB user memory tablespace consists of a list of data pages in the main memory and the checkpoint images on the disk. Unlike disk tablespaces, the storage hierarchy of memory tablespaces is flatter since it does not make use of segments and extents, thus providing much faster performance.

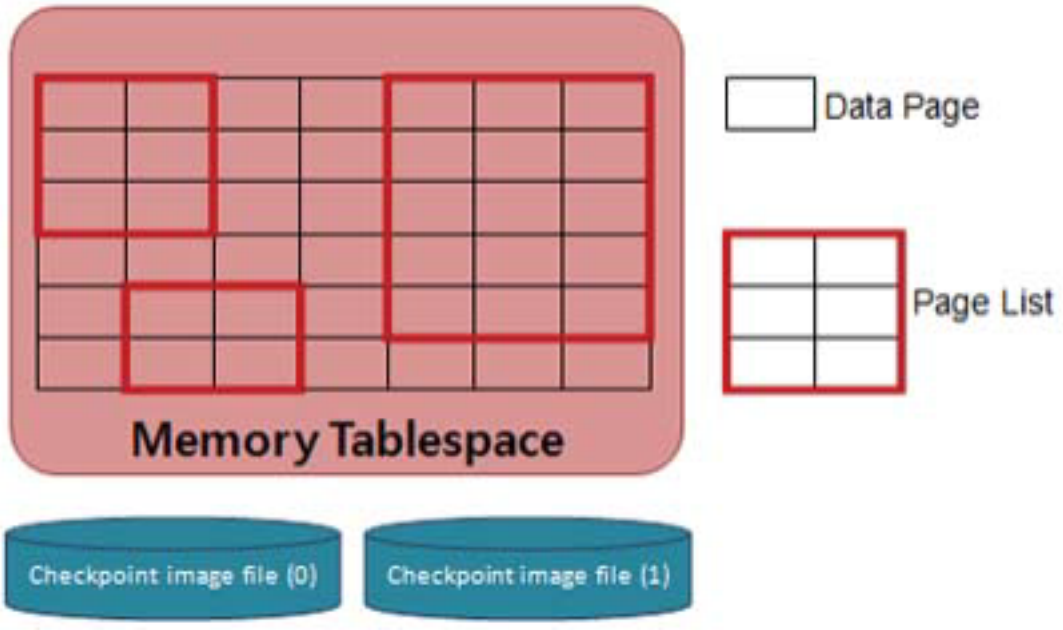

Figure 3 – User Memory Tablespace Structure

The memory data pages are saved logically in the memory tablespace of the system's main memory sequentially. A database table can be thought of as a list of data pages.

Memory Tablespace Indexes

By default, memory table indexes are built in a temporary area of memory (non-persistent index) rather than in user memory tablespaces. In other words, memory table indexes are not stored on the disk area even when system shuts down normally.

Therefore, memory table indexes are always re-created when systems start up. If the size of the database is very large, the index rebuilding process may require some time to complete.

Memory Tablespace Checkpointing

Durability element of ACID refers to the persistence of transactions after they have been committed. As a fully ACID compliant database, to ensure the durability of transactions, ALTIBASE HDB manages transaction logs (redo logs) in which the changes to the database are captured.

A checkpoint refers to the copy of main memory stored on disk. Checkpointing is the action of creating a checkpoint.

The purpose of checkpointing is to write all of the modified pages out to disk from memory tablespace thus providing a physically consistent point for faster recovery.

As part of the recovery process, ALTIBASE HDB uses a combination of transaction log files and checkpoint image files.

When the checkpointing occurs, memory tablespaces are physically backed up to checkpoint image files on normal disk file systems. Although checkpoint image files are not directly required for the normal operation of the database, they are required in order to reduce the recovery time in case of a database restart.

Checkpoint image files contain information about the transactions that are active, their states, and the LSNs (log sequence numbers) of their most recently written logs, and also the

modified pages (dirty pages) that are in the buffer pool.

ALTIBASE HDB makes use of a checkpointing implementation called Fuzzy/Ping-Pong checkpointing. In this implementation, two sets of checkpoint image files are maintained, and used alternately for further safety.

In addition, each checkpoint image can also be divided into several smaller files, with the goal of distributing disk I/O expense across multiple disks.

Checkpoints are performed during system startup, at periodic intervals or by user request.

One of the key benefits of running checkpoints at periodic intervals is to reduce the amount of log data required for the recovery process which in turn helps reduce system downtime in case of a failure. Running checkpoints at periodic intervals can also help prevent the accumulation of transaction log files as they can grow in size to fill up the disk.

Another important thing to keep in mind, in case of corrupt/damaged checkpoint image files, is that the memory objects related to those checkpoint image files cannot be used.

Memory Tablespace Attributes

When an ALTIBASE HDB database instance is created for the very first time, a default system disk tablespace and a default system memory tablespace are automatically created. The internal names of these tablespaces respectively are SYS_TBS_DISK_DATA and SYS_TBS_MEM_DATA.

Both user disk tablespaces and user memory tablespaces are created by the user using the CREATE TABLESPACE statement.

Tablespace attributes that can be specified when a tablespace is created vary depending on whether the tablespace is a disk, memory or volatile tablespace.

Unlike a disk tablespace, in which multiple data files are managed, in a memory tablespace, the objects are stored in a single continuous memory space. Therefore, when a disk tablespace is created, some of the attributes that are specified apply to individual data files, whereas when a memory tablespace is created, all attributes apply to the entire memory tablespace.

When creating a user memory tablespace, the following attributes associated with that memory tablespace should be defined;

User memory tablespace name: User must specify a name for the memory tablespace. The tablespace name must be unique. More than one tablespace having the same name cannot be created. The names of the checkpoint image files are automatically generated based on the user specified tablespace name. Their filename format is Tablespace Name-{Ping Pong Number}-{File Number}'. Ping Pong number is either 0 or 1. As an example, if the name of the memory tablespace is MYMEMTBLSPC, the checkpoint file names will be auto-generated as MYMEMTBLSPC-0-0 and MYMEMTBLSPC-1-0. Because each checkpoint image can be divided up and stored as multiple files, File Number, at the end of the filename, indicates the number of each checkpoint image file, which begins at 0 and increments by 1.

SIZE this is the amount of memory that must be initially allocated when a memory tablespace is created. The size can be specified in kilobytes, megabytes, or gigabytes. If no units are specified, the default unit is kilobytes.

AUTOEXTEND: This determines whether the size of the memory tablespace will be increased automatically. If it is set to ON, the tablespace is automatically increased in size by the system, whereas if it is set to OFF, the user must explicitly increase the size of the tablespace. The extension increment size can be specified by the user

MAXSIZE: This indicates the maximum size to which a memory tablespace can be extended. Like the initial size, it cannot exceed the amount of total memory space available in the system. If it is set to UNLIMITED, the tablespace is automatically increased in size until the total size of all memory tablespaces in the system reaches the limit specified in the MEM_MAX_DB_SIZE system property.

CHECKPOINT PATH: As it is explained earlier, ALTIBASE HDB uses the Ping-Pong checkpointing method for high-performance transaction processing in memory tablespaces. For the Ping-Pong checkpointing, at least two sets of checkpoint images are created on disk. Each checkpoint image can be divided into several files and saved in that form. The size of these files can be specified using the SPLIT EACH clause. These files can be stored in different paths in order to distribute the expense of disk I/O. The user can add or change paths for saving checkpoint image files, but cannot change the size of these files once it has been set.

Memory Tablespace States

User memory tablespaces can be in 3 different states as follows;

- Online
- Offline
- Discarded

Typically, all memory tablespaces operate in online state which is the default state. In this state, all objects and data stored in memory are online.

The state of a memory tablespace can be changed to online, offline or discarded states using the ALTER TABLESPACE command while the database is in the normal operational status (SERVICE) or it is in the META (data dictionary maintenance) status.

This is another area where memory tablespaces differ from volatile tablespaces since the state of a volatile tablespace cannot be changed.

In the offline state, a memory tablespace is not loaded into memory and data objects cannot be accessed. In the offline state, the changed pages are recorded in related checkpoint image files, and the memory which is allocated to this tablespace is freed back to the system.

Memory Tablespace Management

In ALTIBASE HDB, a memory tablespace is structurally different from a disk tablespace. However, it is designed and implemented in a way that enables users to perform identical tasks regardless of the type of tablespace. This is achieved by using the same interface for both memory and disk tablespaces. In other words, users can interact with memory tablespaces just like they do with disk tablespaces.

Some of the common management operations that can be performed by ALTIBASE HDB database administrators on memory tablespaces are as follows:

- Memory object distribution
- Backup and recovery of memory tablespaces
- Managing memory resource limitations
- Disk load balancing

Memory Object Distribution

ALTIBASE HDB memory tablespaces provide the flexibility to database administrators who may want to organize and distribute memory tablespaces according to different criteria such as the importance of the data, or the frequency at which they are accessed.

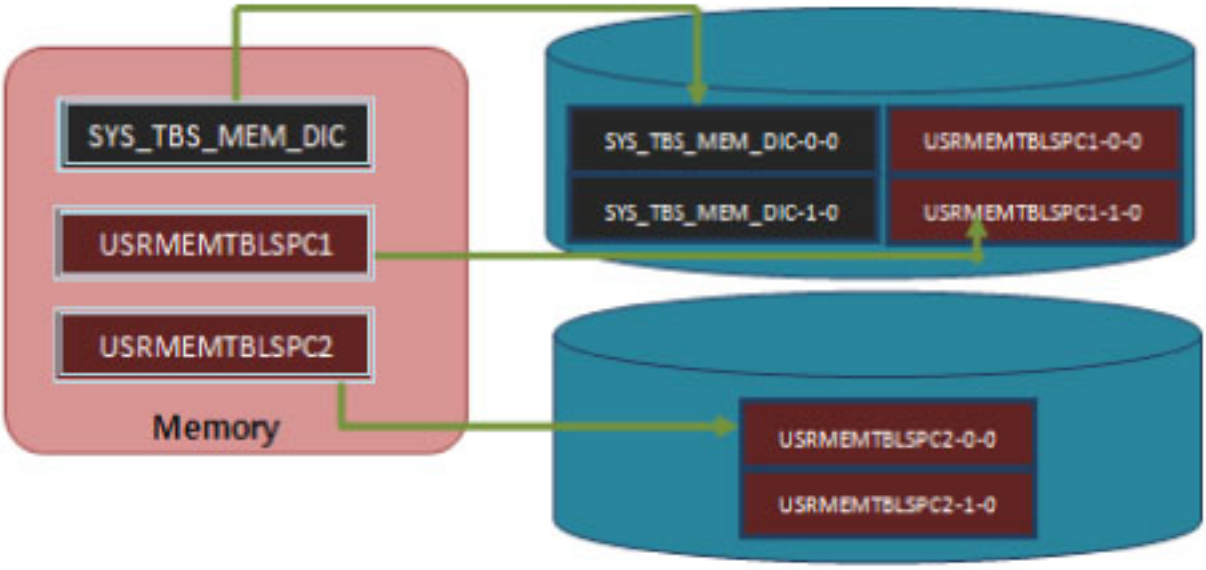

Figure 4 – Memory Tablespace Distribution for Data Protection

When using user memory tablespaces, all objects in memory are managed by using checkpoint image files. In case of corrupt or damaged checkpoint image files, the memory objects related to those checkpoint image files cannot be used. To minimize the risk of losing data, it is recommended that checkpoint image files are distributed across different physical disks as illustrated in Figure 4.

By distributing checkpoint images of memory tablespaces this way, in case of a failure condition, database administrators need to deal only with tablespaces in question rather than worrying about the entire database.

Managing Memory Tablespace Growth

It is also possible to run into problems if the growth of memory data is not well managed. To avoid such possible issues ALTIBASE HDB enables database administrators with the controls to limit the growth of memory tablespaces.

One of the ALTIBASE HDB properties to control the growth of memory tablespaces is the MEM_MAX_DB_SIZE property. This property allows database administrators to specify and limit the maximum size that a memory database can dynamically grow to.

As it is explained earlier, when the AUTOEXTEND attribute is turned on for a memory tablespace, ALTIBASE HDB performs an automatic extension. This operation involves adding up the size of all memory tablespaces and comparing the total size against the size specified in the MEM_MAX_DB_SIZE property.

If the selected extension size is too small, automatic extension process can occur too often. This represents another area of concern for database administrators, since if this operation occurs too often, system performance can be negatively impacted.

Backup and Recovery of Memory Tablespaces

For user memory tablespaces, ALTIBASE HDB supports backup and recovery methods similar those used in support of disk tablespaces. Database administrators have the ability to backup data by tablespace units without differentiating between disk and memory tablespaces. They can perform a full or an incomplete recovery using backed up images.

There are two types of backup modes; Online (Hot backup) and Offline (Cold backup).

Online backup refers to the backup process that is conducted while the database is actively providing service. Because the online backup does not influence the execution of transactions, it can be performed during the service phase.

When performing an online backup, the entire database can be backed up, or just a specific tablespace or a log anchor file as desired.

In the case when the entire database is backed up, because ALTIBASE HDB is a hybrid data-

base, memory tablespaces are always backed up first.

Online backup is only possible when the database is operating in archive log mode. In archive log mode, because all log files are backed up in a separate storage space, a sufficiently large storage space must be set aside, even if checkpointing and log flushing have just been conducted.

After the system is restarted, the most recent image of a memory tablespace can be recovered by repeating all recent transactions using online logs and rolling back all uncommitted transactions using undo logs since some uncommitted data may also have been backed up.

One important factor to keep in mind is that the checkpointing and the online backup are considered mutually exclusive, and cannot be run at the same time.

If a request to perform an online backup is received during checkpointing, the online backup will wait until the current checkpointing is complete. Similarly, if a request to perform a checkpointing is received while an online backup is underway, the checkpointing will wait until the online backup is complete.

When a memory tablespace is backed up offline, the tablespace service is suspended while the backup is performed.

Offline backup is faster than online backup, and enables recovery to be performed more quickly.

Offline backup is only possible when the database is operating in no archive log mode. Offline backup is performed by copying data files, log files and log anchor files after the database is shut down normally.

When a data file is damaged or lost due to a failure in the system, it can be restored only up to the time point at which offline backup was most recently performed.
The Altibase recovery policies support both full (complete) recovery and partial (incomplete) recovery.

Full recovery refers to restoring a data file up to the current time point without losing any online log or archive log.

Incomplete recovery refers to the case in which archive log files or online log files have been lost, and therefore the database is recovered to the point in time immediately before the log files were lost.

Managing Memory Resource Limitations

Because all memory objects need to be loaded into the system memory during a database startup, if there is not enough memory resources to accommodate those objects, service interruptions can occur. To avoid such situations, unneeded data needs to be managed by database administrators.

In order to avoid running out of system memory space, database administrators need to make more memory available to ALTIBASE HDB. This process typically requires a system restart, or the reorganization of the entire database. However, in most cases, this is not desirable because the time required to reorganize the entire database, especially when dealing with massive amounts of data, can be a substantial amount of work, or a restart of the database can cause significant system down times.

ALTIBASE HDB provides an online/offline mechanism for memory tablespaces to help database administrators manage such situations easier. Database administrators can put memory tablespaces into offline state in order to reduce unnecessary data while still providing uninterrupted services to users. When database administrators are done with the purge process, they can bring memory tables back to online state again.

When the database is running, it is possible for a particular user memory tablespace to occupy more memory than it actually requires. This can often happen as a result of an update or a delete operation on the existing data.

In such cases, for efficient use of memory resources, the unneeded memory should be returned back to the system. To do that, ALTIBASE HDB provides database administrators with a memory table compaction function.

ALTER TABLE statement with the use of the COMPACT syntax returns the free pages in a memory table or a volatile table back to the system.

In addition, the TABLE_COMPACT_AT_SHUTDOWN property is provided to automate the

compaction process during server shutdown. If the value of this property is set to 1, during shutdown table compaction is automatically performed on the user memory tables.

Discarded state is a stage used for deleting user memory tablespaces. Once a memory tablespace is in this state, it cannot be changed to any other state. A typical use for this state is when the database cannot be started due to a corrupt checkpoint image file. Once the data is recovered and the corrupt tablespaces are deleted, the database can operate normally again.

Disk Load Balancing Considerations

When multiple physical disks are available in the system, database administrators should consider distributing transaction logs, system tablespaces and user tablespaces across the available disks to load balance the disk I/O.

As it is explained in the earlier sections, the specific implementation of Fuzzy and Ping-Pong checkpointing techniques in ALTIBASE HDB does not impact the performance of database transactions. However, in the mix use environment, where disk tablespaces are used in combination with memory tablespaces, it is possible to have I/O contention between disk tablespace and memory tablespace operations which can impact the overall database performance.

Whenever possible, the separation of storage of transaction logs, system tablespaces, user disk tablespaces and checkpoint image files of user memory tablespaces over multiple disks is recommended as illustrated in Figure 5.

Memory Tablespace Monitoring

Database administrators can view memory tablespace information through the performance views that ALTIBASE HDB provides.

The names of some of the key performance views related to ALTIBSASE HDB tablespaces are:

V$TABLESPACES
V$MEM_TABLESPACES
V$MEM_TABLESPACE_CHECKPOINT_PATHS
V$MEM_TABLESPACE_STATUS_DESC

Database administrators can determine the type and status of each tablespace by using V$TABLESPACES performance view after starting up the database at Control Level.

When ALTIBASE HDB is in service, database administrators can view the properties of user memory tablespaces such as size and usage of memory tablespaces utilizing the V$MEM_TABLESPACES performance view.

Database administrators can also confirm specific status of memory tablespaces through the V$MEM_TABLESPACE_STATUS_DESC performance view.

V$MEM_TABLESPACE_CHECKPOINT_PATHS performance view allows database administrators to view specified checkpoint image paths of memory tablespaces.

V$MEMTBL_INFO performance view provides information about memory tables.

V$MEM_BTREE_HEADER and V$MEM_BTREE_NODEPOOL performance views provide index information generated in memory tables.

It is also possible to perform join operations between these performance views.

Company Overview

ALTIBASE is a leading provider of data performance solutions that provide real-time access, analysis, and distribution of high volumes of data in mission critical environments.

As the importance of real-time information grows, ALTIBASE data performance solutions deliver real-time access to your data whether your data is at-rest in a database or it is still in-motion across the wire.

ALTIBASE helps its customers maximize their data investments by providing real-time data performance solutions. In today's competitive business environment, ALTIBASE enables companies to drastically improve the speed of data access and analysis across the enterprise.

ALTIBASE delivers its data performance solutions through two key products:

• ALTIBASE® HDB™ is a hybrid relational DBMS that combines in-memory and on-disk storage in a single relational database.

• ALTIBASE® DSM™ is a data event middleware that filters, analyzes and distributes high-volume data streams in real time.

ALTIBASE