eXtremeDB™
Transaction Logging Edition

Combine the performance of an in-memory database with the persistence of a disk-based database.

"With a 10X increase in performance, eXtremeDB-TL eliminates disk I/O as the overwhelming performance bottleneck."

-- Object Computing, Inc.

*eXtreme*DB, the real-time embedded database for devices that are eXtremely innovative

Overview

Transaction logging provides recovery capabilities for *eXtreme*DB databases in the event of device or system failure. When transaction logging is active and enabled for the database, all updates to the data objects are logged to a set of files on disk, or a network device. If the memory content is damaged or destroyed, the exact state of the database is restored by an automatic roll-forward procedure.

The Challenge

In-memory databases deliver exceptional performance via all-in-memory processing. To meet user needs for fault-tolerance and availability, these databases must provide a way to safeguard in-memory data without sacrificing performance and efficiency. McObject meets this challenge with an array of *eXtreme*DB product features and extensions.

First, every version of *eXtreme*DB has the backup and restore capability to periodically stream an in-memory database image to some external endpoint, such as a local or network disk file.

Other solutions include using non-volatile RAM that retains the data in the case of system failure, or building in high availability through a replication scheme.

However, a non-volatile memory device is not practical for all systems, and frequent online database backups or the interprocess communication incurred with high availability implementations can decrease overall system performance.

For this reason, McObject offers the **eXtremeDB Transaction Logging** edition of its in-memory embedded database. Transaction logging writes database changes into a transaction log on persistent media, combining the performance and efficiency benefits of an in-memory database with the persistence of a conventional disk-based database.

Transaction Logging Implementation

Transaction logging does not alter the all-in-memory architecture that provides *eXtreme*DB with striking performance advantages over disk-based databases. Read performance is

unaffected by transaction logging. Write performance still far exceeds that of disk-based databases for the simple reason that while *eXtreme*DB transaction logging requires exactly one write to the file system per database transaction, a disk-based database transaction requires many writes to update data pages, index pages, transaction log, etc. For the disk-based DBMS, the larger the transaction, the more writes that are necessary.
To minimize performance impact, *eXtreme*DB equips developers with highly flexible tools for tuning *eXtreme*DB transaction logging in applications.
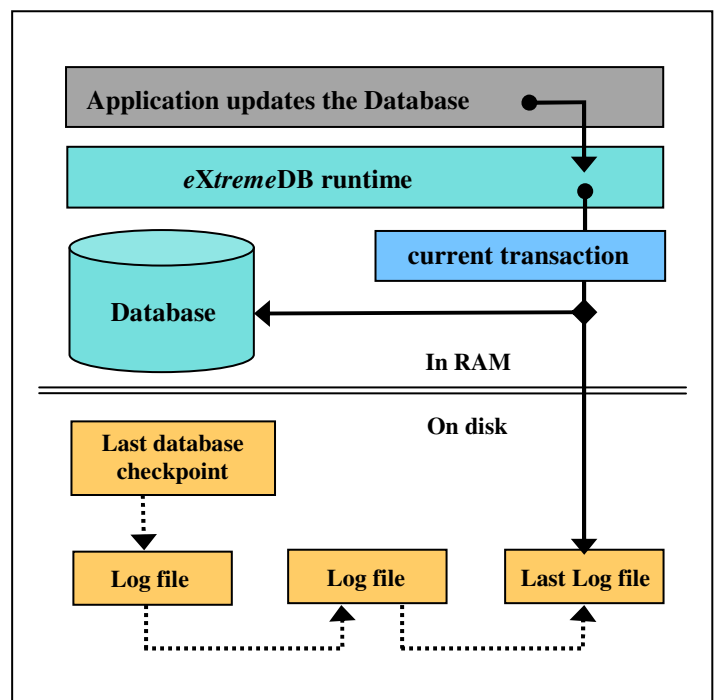
Logging may be set to different levels of transaction durability. Cleanup of outdated logs and database images ("checkpoints") can be synchronized with checkpoint completion, or delegated to a background task. To provide the greatest flexibility, McObject provides a rich array of transaction logging control parameters, including depth of the recovery tree and maximum transaction log file size.

Pre°ision Data Management