



# Oracle8i™ High Availability Enhancements

Features Overview

November 1998

## PRODUCT SUMMARY

Oracle8i merges the data management power of the world's leading database with the collaborative capabilities of the Internet. The Internet extends the reach of database applications throughout organizations and communities. This reach further highlights the importance of high availability in data management solutions. Small businesses and global enterprises alike have customers all over the world requiring access to data 24-hours per day. Recognizing these demands, Oracle has made extending its industry-leading database availability and reliability one of Oracle8i's primary goals. Oracle8i contains a significant number of enhancements and new features designed to provide highly-available database solutions. These improvements go beyond making the Oracle database server more reliable. They enhance system fault recoverability, allow planned database maintenance while the database is in production and available to users, improve system manageability and serviceability, and simplify disaster planning. Together, these new availability features will improve an administrator's ability to support users and meet service level objectives. Most importantly, since Oracle8i runs everywhere, high availability solutions can be implemented within workgroups, departments, the data center, as well as the largest worldwide enterprises.

## SYSTEM FAULT RECOVERY

System faults such as hardware, operating system, or database failure are potentially very disruptive as they lead to unplanned system downtime. Oracle8i's comprehensive availability architecture includes a robust set of features to quickly and automatically recover from system faults with minimal disruption to end users.

	Level	Oracle8i Feature	
↑ Increasing Availability	Hot Failover	Preconnect	Transparent Application Failover (TAF)
	Warm Failover	Clustering with Parallel Server	
	Cold Failover	Clustering	
	Restart	Fast-Start Fault Recovery	

Figure 1: Oracle8i Availability Hierarchy

Oracle8i provides an availability hierarchy, with increasing levels to meet the diverse needs of today's organizations. Oracle8i's Transparent Application Failover can complement all levels of the availability hierarchy to mask failures from end users. The levels of the availability hierarchy are:

- *Restart* - Oracle8i's Fast-Start Fault Recovery provides fast and predictable recovery from system failures.
- *Cold Failover* - Oracle8i can be restarted on another system in a cluster. This allows the application to be quickly restarted, without the delay of a repair and reboot.
- *Warm Failover* - Clients will failover to a peer system in a cluster. The peer will already have the database mounted using Oracle8i's Parallel Server. This saves time when restarting the server, and if the peer instance has been active, its pre-warmed cache may provide better performance.
- *Hot Failover* - Clients will failover to a pre-connected peer system. This reduces the time and complexity for users to reconnect, which can be several minutes if all users try to reconnect at once after a failure.

### **Fast-Start Fault Recovery**

One of the most common causes of application downtime is a system fault or crash. System faults are the result of hardware failures, power failures, and operating system or server crashes. One of the primary goals of Oracle8i is to make system fault recovery as near instantaneous as possible.

Not only is Oracle8i system fault recovery extremely fast, but it can also be bounded by the database administrator, making recovery time predictable, and improving the ability to meet service level objectives.

### **Conventional Checkpointing Architecture**

After a system fault, Oracle automatically recovers business data to a consistent state by performing fault recovery. The conventional way a database recovers from a system fault is to perform two critical tasks:

- Ensure that all changes made to the database before the system fault are captured and applied. This is called the *roll-forward* phase of fault recovery. The work required to do roll-forward processing is proportional to the rate of change to the database (update transactions per second) and the time between which consistent snapshots, or *checkpoints*, of the database are made.
- Ensure that all uncommitted changes to the database are undone. This is called the *roll-back* phase of fault recovery. The work required to do roll back is proportional to the number and size of uncommitted transactions when the system fault occurred. Thus, the time to do roll back is not under the control of the administrator.

The total recovery time is the sum of time to do roll forward, plus the time to do roll back.

The fault recovery process reapplies modifications made to the database since the last checkpoint. Conventional checkpoint algorithms periodically identify all the modified blocks in the buffer cache and write them in bulk to the database. After all the writes are completed, the checkpoint time (i.e., the point from which recovery would have to start) is advanced to the time the checkpoint started. This results in an automatic built-in delay in database recovery since all update transactions during one, and possibly two checkpoint intervals, will have to be recovered after a system fault.

Oracle8i introduces a new recovery architecture called *fast-start fault recovery*. Fast-start fault recovery reduces the time required for roll forward and makes it bounded and predictable. Fast-start recovery also completely eliminates roll-back time for transactions aborted due to system faults. So in Oracle8i, total recovery time after a system fault is significantly reduced because roll-forward time is minimized and roll-back time is eliminated.

### Fast-Start Checkpointing

The basis for fast-start fault recovery is Oracle8i's new fast-start checkpointing architecture. Instead of doing periodic checkpointing, as is conventionally done, fast-start checkpointing occurs continuously, advancing the checkpoint time as blocks are written. Fast-start checkpointing always writes the oldest modified block first, ensuring that every write allows the checkpoint time to be advanced. This eliminates bulk writes, and the resultant I/O spikes that occur with conventional checkpointing, yielding smooth and fast ongoing performance. Continuous advancement reduces roll forward by half compared to conventional checkpoints at the same transaction rate.

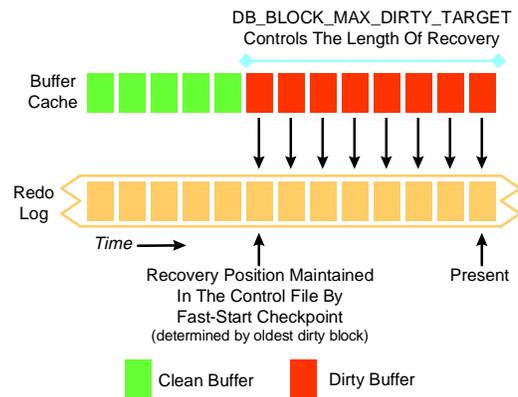


Figure 2: Fast-Start Checkpointing

Figure 2 shows how Oracle automatically varies the rate of checkpoint writes to meet a user-specified roll-forward target. The administrator specifies a limit, or bound, on how long roll forward will take instead of how often checkpoints should occur. The Oracle server automatically adjusts the checkpoint write rate to meet the specified roll-forward bound while issuing the minimum number of writes. Conventional checkpoints must be set to handle the highest possible modification rate, and therefore, perform several times too many writes during normal system loads.

### **Fast-Start Roll Back**

The roll-back phase of system fault recovery in Oracle8i has been eliminated using a technology called non-blocking roll back. New transactions can begin immediately after roll forward completes. When a new transaction accesses a row locked by a dead transaction, the new transaction will roll back only the changes that block its progress. This can be done extremely efficiently because it uses the same technology as Oracle's unique multi-version read consistency. New transactions do not have to wait for the entire dead transaction to be rolled back, so long-running transactions no longer affect recovery time. The fast-start technology maximizes data availability and ensures predictable recovery time.

In addition, the database server now has the ability to roll back dead transactions in parallel. This technique is used against rows not blocking new transactions, and only when the cost of performing dead transaction roll back in parallel is less than performing it serially.

In summary:

- The fast-start fault recovery architecture provides rapid and predictable recovery from system faults.
- Roll forward is made smooth, rapid, and predictable.
- Roll back has been eliminated and no longer blocks new transactions.
- Administrators are fully in control of recovery time.

### **Clustering for High Availability**

Clustering two or more systems is a common way to provide a highly available server platform. A properly implemented cluster removes the host system as a single point of failure from the solution. After a system fault, Oracle8i can failover to a spare system in the cluster. This cold failover removes the time to repair and restart the primary system from recovery time. Most systems vendors offer products to control cluster failover of Oracle8i. On Windows NT systems, Oracle offers Oracle Fail Safe to control failover of an Oracle8i database within a cluster.

The Parallel Server option for Oracle8i provides users greater levels of availability and scalability. In the event of a system fault in the cluster, Oracle8i continues to provide data access from peer systems in the cluster. This warm failover removes from recovery the time to repair and reboot the failed system and restart the database.

Pre-connecting users to a primary and secondary Oracle instance in a Parallel Server cluster can further improve recovery. With hot failover, users pre-establish connections with the secondary instance on the peer system. In the event of a failure on the primary server, users activate the pre-established connection on the peer system, which can save significant time in the recovery process.

## **Transparent Application Failover**

Fast-start fault recovery and clustering allow the Oracle server to quickly recover from system faults and minimize any impact on the end user or application. Oracle8i with Transparent Application Failover (TAF) can further mask failures from applications. TAF provides an application the ability to automatically reconnect if the connection to the database is broken. Client sessions do not need to manually reconnect after an instance crash, as the server will automatically reestablish those connections. Furthermore, any queries in process at the time of the failure can automatically be restarted, and data returned to the application will resume from the point the failure occurred.

Transparent Application Failover supports failover to another server, or recovery by a stand-alone server. It therefore can be used in a wide variety of implementations, including Oracle® Parallel Server, Oracle® Fail Safe, systems-vendor cluster failover solutions, replicated databases, standby databases, and stand-alone database servers.

## **ONLINE REORGANIZATION AND DEFRAGMENTATION**

The Internet has made it easy to share data globally, but brings new challenges and requirements for data availability. As global users access data 24-hours per day, maintenance windows have all but evaporated. Planned downtime is becoming as disruptive as unplanned downtime, as there are no longer any windows of time where users will not be affected. When the volume of data stored in a database becomes very large, maintenance operations can be quite time consuming. It is important that these operations be performed without impacting users of the data.

Oracle8™ introduced partitioned tables and indexes, which allowed administrators to perform maintenance on these objects, one partition at a time, while the other partitions remained online. This minimizes the impact of maintenance activities on data availability. Oracle8i improves on this with new features that allow administrators to perform required maintenance while tables and indexes are completely online and fully available to end users. These new online Oracle8i maintenance capabilities improve data availability, query performance, response time, and disk space utilization, which are all important to the 7x24 Internet enterprise operation.

### **Online Index Creation, Rebuild, and Defragmentation**

Oracle8i can create an index online, allowing users to update and query the base table while creating the index. No table or row locks are used during the creation operation. Changes to the base table and index during the build are recorded in a journal table, and merged into the new index at the completion of the operation. These online operations also support parallel index creation, and can act on some or all of the partitions of a partitioned index. Online index creation and rebuild improves database availability by providing users full access to data in the base table during an index build.

Online index rebuild and defragmentation is important to maintain performance. After numerous inserts and deletes, an index can become fragmented, wasting space and impacting performance. Oracle8i provides two facilities to defragment an existing index: online rebuild and in-place coalesce. While the index is online, Oracle8i can rebuild the index completely to recover space and relocate the index. As with index creation, Oracle uses a journal to record changes made to the base table during the rebuild process. If storage space for rebuilding the index is limited, Oracle8i can also perform online in-place coalescing of an index to defragment the index.

### **Online Table Move and Reorganization**

Oracle8i allows Index-Organized Tables (IOTs) to be moved and reorganized while users are accessing and updating the data in the table. Changes to the table during the operation are recorded in a journal table and merged with the table at the completion of the operation. These online operations can be performed on a complete IOT or on a single partition in a partitioned IOT. This new online capability allows administrators to physically move an IOT with no impact to end users of the data in the table.

As with table indexes, numerous insert and delete operations can potentially cause an IOT to become fragmented, wasting space and impacting performance. Online rebuild of IOTs reduces fragmentation and frees up space, resulting in improved storage utilization and query performance. Performing this operation online ensures that users will continue to receive both optimal performance and continuous data availability.

IOTs have been enhanced to use logical ROWIDs. Logical ROWIDs contain the primary key of the referenced row, instead of a physical location within the database. This means that if a row is moved to a new location, the logical ROWID will still contain a valid reference to the row. To retain the speed of conventional ROWIDs, logical ROWIDs also contain a physical location hint which can be used to optimize performance. If an index is created on a field other than the primary key, Oracle8i will create a logical index using logical ROWIDs to reference the table rows. If a table row is moved to a new location, the logical index will not need to be updated.

With the release of Oracle8i, IOTs now support all the features of conventional tables, and provide the added benefit of online reorganization and move. Because of the availability and manageability advantages of IOTs, Oracle recommends the index-organized option be used for all new tables that may become fragmented or require reorganization.

### **Online Read-Only Tablespaces**

Read-only tablespaces allow data files to be placed on read-only media such as CD-ROMs and eliminate the need to perform backups of static portions of the database. Oracle8i lets an administrator make a tablespace read-only while the system is in production and actively accessed by users. To maintain consistency, Oracle8i waits for transactions started before the command was issued to end before making the tablespace read-only. New transactions are prevented from updating the tablespace during this period, but concurrent updates against other tablespaces are allowed. The ability to make a tablespace read-only while the database remains online increases the availability of sites that use read-only tablespaces.

### **SYSTEM MANAGEABILITY AND SERVICEABILITY**

The ability to easily and accurately perform system management tasks is important to maintaining overall database availability. Oracle8i provides powerful tools to manage and simplify management. Oracle8i also provides serviceability tools to help administrators quickly recover from errors should they occur.

## **LogMiner™ SQL-Based Log Analyzer**

Oracle log files contain a wealth of useful information about the activities and history of an Oracle database. Until Oracle8i, there was no easy to use tool that could tap into this information. LogMiner is a fully relational tool which allows redo log files to be read, analyzed, and interpreted by the user using SQL. LogMiner can view any valid redo log file (online or archived) from Oracle release 8.0 forward. Log files contain all of the data needed to perform database recovery. They also record every change made to data and metadata in the database. Analysis of the log files with LogMiner can be used to accomplish the following:

- Track specific sets of changes based on transaction, user, table, time, etc. It is easy to determine who modified a database object and see the before and after data. Tracing and auditing database changes back to their source, and undoing them if necessary, are valuable security and management tools.
- Pinpoint when an incorrect modification is introduced into the database. This can be used to perform logical recovery at the application level, instead of at the database level.
- Provide supplemental information for tuning and capacity planning. Administrators can also perform various forms of historical analysis to determine trends and data access patterns.
- Retrieve critical information for debugging complex applications.

No additional collection overhead is incurred to obtain information from LogMiner. The log files to be analyzed are mapped to a dynamic performance view (V\$ table) and are then accessed via standard SQL, or programmatically from any supported language/interface (PL/SQL™, Oracle® Call Interface, Pro\*C™, Java Stored Procedures, SQLJ). Each row in the log table represents complete information on a logical operation performed in the database. For each of these logical operations, REDO and UNDO SQL statements that logically describe the effect of the change are provided. LogMiner can be used as the underlying technology for a powerful set of management and analysis tools.

## **Database Resource Manager**

The functionality and performance of the database and applications are important components of their availability. If the database is available, but users are not achieving the level of performance they need, then availability and service level targets are not being met. Database manageability and availability are impacted by how resources are distributed to the various applications accessing the database. In Oracle8i, a sophisticated database resource manager can be used to allocate and manage CPU resources among users and applications. It also allows the administrator to limit the degree of parallelism of any operation—making it possible to balance one user's resource consumption against another's, and to partition system resources among tasks of varying importance to achieve overall enterprise goals.

Resources are allocated to users based on a plan that is specified by the database administrator. Users, and groups of users, are assigned to “resource consumer groups.” A “resource plan” specifies how resources are distributed among the different resource consumer groups. Plans can be dynamically altered on a production database without quiescing the system. This allows alternate plans for day time, night time, weekends, quarter end, or other times that require a different set of priorities. The resource manager is fully integrated into the database security system. Users, or their sessions, can switch resource consumer groups to increase execution priority, if the user has been granted the privilege to switch consumer groups. Users can also be moved from group to group by the administrator on a production system, thereby dynamically changing how CPU resources are used. These capabilities allow very simple, yet very powerful allocation policies to be implemented for database resources. The following example illustrates some of the capabilities of the resource manager.

Company XYZ sells widgets directly to customers over the Internet. In order to assure optimum performance for customers placing orders, 80 percent of the CPU is allocated to service customer orders. The remainder of the CPU resource is divided between the processing required to ship the widgets and that required to bill for them.

Company XYZ Resource Plan		
Resource Consumer Groups	Level 1 CPU Allocation	Level 2 CPU Allocation
Web-based Order Entry	CPU Allocation = 80%	CPU Allocation = 0%
Shipping	CPU Allocation = 0%	CPU Allocation = 75%
Billing	CPU Allocation = 0%	CPU Allocation = 25%

Unused CPU resources allocated for order entry are made available to do shipping and billing, but order processing will be guaranteed its allocation if the order rate is high enough to require these resources. Unlike conventional priority schemes, shipping and billing will receive their allotment of resources, if required, and cannot be starved by high demand for resources by the Order Entry consumer group. If necessary, Company XYZ can dynamically alter its resource plan to increase the CPU resource allocated for billing. Database resource management policies can be implemented to guarantee measured database service to the applications in support of changing business needs.

## Transportable Tablespaces

Transportable Tablespaces is a new feature in Oracle8i that allows a user to move or copy a subset of one Oracle8i database to another Oracle8i database. Transporting a tablespace from one database to another can be orders-of-magnitude faster than either export/import or unload/load because it involves only copying data files and integrating metadata. Applications for this technology include:

- Moving data from an operational OLTP system into a data warehouse. Since the amount of data in a data warehouse is very large, this could greatly speed the process and make the warehouse more available.
- Publishing structured data for incorporation into remote databases. For example, a retailer might want to distribute to its stores an updated parts catalog each month. A tablespace containing the catalog could be created at headquarters and placed on CD-ROM. This would be sent to all the stores where it could easily and quickly be integrated into each local store's system.
- Performing tablespace point-in-time recovery. By transporting a good backup copy of the tablespace into the production database, service can be quickly restored improving database availability.
- Offloading of CPU-intensive data load operations to a staging database.

Transporting a tablespace is fast and simple. First, the tablespace is put into read-only mode to ensure that a consistent image of the data is captured. Then metadata describing the tablespace is exported from the source database dictionary. This export is very fast because the data exported is very small. Next the tablespace files are copied to the destination database using any operating system facilities (copy command, ftp, etc.). At the destination, the metadata describing the tablespace is imported. Again, this is a very fast because the size of the import is very small. Finally, if desired, the imported tablespace can be placed in read-write mode.

An additional benefit of transporting tablespaces is that indexes can be transported as well as table data. This can save a great deal of processing for a large table.

## Locally-Managed Tablespaces

Database administrators are concerned about extent fragmentation in tablespaces, as fragmentation can reduce space utilization and database performance. Prior to Oracle8i, management of free and used extents relied heavily on the data dictionary tables. Oracle8i introduces Locally-Managed Tablespaces with an improved space management implementation. This new tablespace type automatically sizes all new extents at standard sizes, improving space allocation performance and eliminating free extent fragmentation issues.

Oracle sizes new extents using either a uniform or automatic policy. If the tablespace uses the uniform policy, all extents are the same size, which is set at tablespace creation time. Since all extents are the same size, no fragmentation can occur. If the tablespace uses the automatic policy, there are only small and large extents. Small extents are automatically used for small database objects, and large extents are used for large objects. Small extents are packed together in a group the size of a large extent. By separating small and large extents, fragmentation is avoided.

The new type of tablespace is called Locally Managed because all extent information is tracked in the tablespace itself using bitmaps. Bitmaps manage space allocation very efficiently, and require no dictionary access and update to allocate an extent to an object. This minimizes access to the data dictionary, improving availability. Because of these improvements, Oracle recommends using Locally-Managed Tablespaces for all new tablespaces.

### **Recovery Manager**

Data backups are critical to database recovery. However, very large database systems impose stringent requirements for backup and recovery operations. In many cases, the physical action of copying the database files can take many hours, due to the large size and number of files. Additionally, backup and recovery operations must be performed with minimal interruption to business processing.

The Oracle8i Recovery Manager provides an integrated method for creating, managing, and restoring backups of a database. Recovery Manager provides greater ease of management and administration of the backup and recovery operations, while maintaining superior performance and increased availability of the database. Recovery Manager can perform a full or incremental backup of an entire database, or a subset, while the database is open or closed. Oracle8i's backup accelerator provides a media management API allowing Recovery Manager to tightly integrate with third-party media management products. These products can now directly copy files at the storage subsystem level, thereby improving performance and reducing the load on the database server.

Also new to Oracle8i is tighter integration in a cluster environment. Recovery Manager will utilize knowledge of file affinity to efficiently backup data in a distributed storage environment. In addition, Recovery Manager can automatically multiplex backups, to protect data from backup media failures.

### **Detecting and Repairing Data Corruptions**

Block corruptions are very rare in Oracle, but if they occur, specialized tools may be needed to repair the database and make it usable. In the past, administrators encountering corrupt blocks had to drop and recreate the object, or request the assistance of Oracle's World-Wide Customer Support (WWCS). Dropping and recreating an object is not always viable, and can be very time consuming. Oracle8i includes a new package, DBMS\_REPAIR, which helps detect and repair block corruption on tables, indexes and other objects. If possible, DBMS\_REPAIR will detect and report corruptions, make corrupt objects usable, repair corruptions, and report any lost data. These tools let the administrator fix corruption problems and get the database back online with a minimum of downtime.

### **Application Change Management**

Every SQL statement submitted to the database is processed by a component of the database server called the optimizer. The optimizer constructs an execution plan that specifies the most efficient way the requested data can be retrieved from the database. Part of this analysis is based on statistics that describe the underlying data in the table and indexes. After upgrading the database or application, the optimizer may generate a different execution plan than was used in the past. Queries running with these new execution plans may exhibit different performance than they exhibited before the change was introduced. This effectively reduces the predictability and increases the risk associated with database and application upgrades and could inhibit taking advantage of new features in the database.

Oracle8i provides two new features that address this. First, the ability to store existing execution plans and insulate an application from upgrades is provided. Plans can remain consistent across ANALYZE commands, creation of new indexes, new releases of Oracle, and other changes. The optimizer uses the stored plan to create an equivalent execution plan for the query. Multiple stored plans can be created for experimentation and a phased roll out of an application can be done with more predictable results. Oracle8i also has enhanced the collection and management of the statistics used to generate execution plans. Statistics can be automatically generated as an index is created with minimal overhead. Statistics gathering can also now be done in parallel. Statistics may be extracted from the dictionary and stored in a user table. They can then be modified and reinserted into the dictionary. This can be used to transfer statistics to or from a test database or to experiment with different versions of dictionary statistics. With Oracle8i, it is now possible to upgrade applications and Oracle releases with more predictability, improving the performance and availability of applications.

## **DISASTER PLANNING**

Disaster planning provides protection from catastrophic events that cripple processing at a site for an extended period of time. Examples include natural disasters, power and communication outages, and even terrorism. Oracle8i offers a variety of disaster recovery solutions that provide the ability to create and maintain a remote copy of a production database at an alternate or primary site. In the event of a disaster, users of the data can continue to function by accessing the remote database.

Oracle supports three disaster recovery solutions: Standby Database, Replication, and Remote Storage Mirroring. Standby Database and Replication have been enhanced with Oracle8i.

### **Automated Standby Database**

Automated Standby Database provides a means to create and maintain a remote copy of a production database. The Standby Database can reside in the same or different data center and take over processing from the primary production database providing near continuous database availability. Oracle has provided the standby database feature for a number of years, but the technology is significantly extended in Oracle8i.

A standby database is initially created by copying or cloning the production database. As archived redo logs are generated on the production database they are applied to the standby database. This allows the standby database to remain synchronized with the production database. Prior to Oracle8i, the archived redo logs had to be manually transported or copied to the standby database and manually applied. With the new Automated Standby Database in Oracle8i, the archived redo logs can be automatically transferred and applied. This eliminates the need for manual procedures to copy and transmit the redo logs and the need for the operator at the backup site to manually specify which logs to apply. Automating this process will eliminate a potential source of human error and increase database and application availability. If desired, multiple Automated Standby Databases of one production database can be created and automatically maintained using this technology providing additional insurance against a data center disaster. Another improvement offered by Oracle8i is the ability of the administrator to create multiple ARCHiver processes. Redo logs entries can be generated faster than the ARCH process can archive them. Running multiple ARCH processes will eliminate archiving as a bottleneck in sites archiving redo logs, such as those implementing a Standby Database.

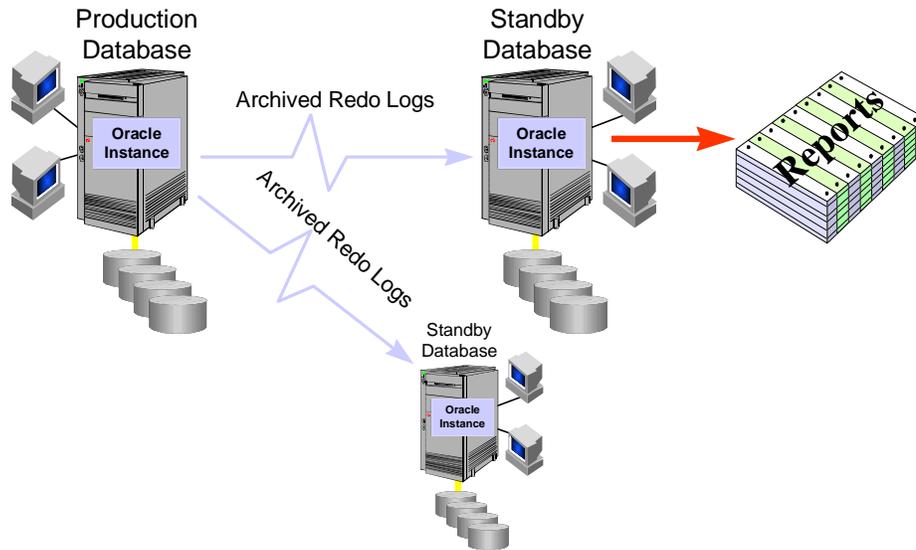


Figure 3: Standby Database

Another new capability of the Automated Standby Database is the ability to use a standby database as a read-only database. In the past a standby database could not be opened or used without subsequently re-cloning the production database. Thus resources allocated to a standby database were idle and could not be used for anything other than backing up the production database. In Oracle8i, the standby database can be taken out of recovery mode and opened for read-only query processing. This can be used to temporarily offload query processing from the production database. When the query processing is complete, the standby database can return to standby mode without re-cloning the production database. Other backup copies of a database can also be opened in read-only mode and used for querying. This provides more productive resource usage while maintaining high database availability.

### Advanced Replication

Another solution for disaster planning is Oracle8i Advanced Replication. Replication is the process of copying and maintaining database objects in multiple databases that make up a distributed database system. Oracle Advanced Replication allows an application to update any replicas of the database, and have those changes automatically propagate to other databases while ensuring global transactional consistency and data integrity. Users of the data can connect to any server and modify the data. Oracle provides sophisticated customizable data conflict resolution methods to automatically avoid, or detect and resolve conflicts when data is simultaneously updated in multiple master databases. Alternatively, administrators may designate one site a master for updates, and other sites read-only for reporting activities.

In the event of a disaster at one of the sites, the surviving databases will remain online. If transactions are continuously propagated between systems, the window of potential data loss is minimized to a very short period of time.

Advanced Replication has been enhanced in Oracle8i to improve performance. New packages have been internalized into the database to increase replication throughput and reduce system processing overhead. This improved replication capacity makes replication a more viable disaster planning solution for systems with high transaction throughput.

## **SUMMARY**

Oracle8i provides breakthrough enhancements to database availability. Its availability architecture provides fast recovery from faults, allows routine maintenance to be performed online, includes an array of system management tools to prevent disruptions, and supports disaster planning solutions to protect from catastrophic events. Because these features are available to every Oracle8i customer, small workgroups and global enterprises alike will be able to extend the reach of their critical business applications. With Oracle8i and the Internet, data is now accessible everywhere, at any time.



**Oracle Corporation**  
**World Headquarters**  
**500 Oracle Parkway**  
**Redwood Shores, CA 94065**  
**U.S.A.**

**Worldwide Inquiries:**  
**+1.650.506.7000**  
**Fax +1.650.506.7200**  
**<http://www.oracle.com/>**

**Copyright © Oracle Corporation 1998**  
**All Rights Reserved**

**This document is provided for informational purposes only, and the information herein is subject to change without notice. Please report any errors herein to Oracle Corporation. Oracle Corporation does not provide any warranties covering and specifically disclaims any liability in connection with this document.**

**Oracle is a registered trademark, and Enabling the Information Age, Oracle8, Oracle8i, LogMiner, Pro\*C, and PL/SQL are trademarks of Oracle Corporation.**

**All other company and product names mentioned are used for identification purposes only and may be trademarks of their respective owners.**