Informix SOFTWARE

Informix Dynamic Server

High Availability for Mission-Critical Applications

Abstract

Informix® offers a variety of high-availability features for critical applications that require 24x7 access to all types of data. Providing this high level of availability requires hardware and software that ensures continuous database availability during maintenance and administration, as well as in the event of network failure. Informix Dynamic Server™ (IDS) high-availability features include online utilities for backup and recovery, reorganization of tables, enterprise replication, cluster and data failover capabilities, software mirroring, and more. These features enable IDS to provide a highly available environment for all types of data processing.

Table of Contents

Market Overview

Around-the-clock database availability is more critical in today's data-driven business environment than ever before. Mission-critical database applications, such as supply chain, order processing, and distribution processes require nearly 100-percent availability, or zero downtime. Even a short amount of downtime can jeopardize the success of a business.

While other applications, such as telecommunication hubs, international banking, and hotel reservation systems, do not have life-or-death consequences, they do require around-the-clock availability. Downtime can delay information for critical business decisions, which can result in lost sales opportunities and tarnished reputations.

Technology Overview

High-availability processing solutions are used for all types of information processing, including online transaction processing (OLTP) and decision support (DSS) processing.

Online transaction processing

Database availability is of critical importance for OLTP systems that receive continuous transaction data. If the database is unavailable because of a hardware or software failure, a business cannot function normally, which can have devastating consequences. Most airline reservation systems, for example, are available 24x7 via a toll-free number to make or confirm reservations. Today, OLTP systems enable passengers to bypass reservation agents and access an airline's Web site to check seating availability, select seats, and process reservation requests. When the airline's OLTP system is unavailable, it can result in loss of revenue and dissatisfied customers who can easily turn to a competitor's Web site for similar services.

As businesses move from traditional OLTP to Web-based OLTP environments, system and database availability become even more critical. With traditional OLTP environments, the numbers of users and transactions are fairly predictable, so administration and maintenance downtime can be planned during slow periods. And because users—typically customer service agents—are employees who have been notified of the downtime, they are generally more tolerant of the downtime.

With Web-based OLTP systems, however, both the number of users and transaction loads are less predictable, making administrative and maintenance operation downtime difficult to schedule. Because Web-based applications can be accessed by huge numbers of users who expect the system to be available 24x7, unexpected downtime on a Web-based OLTP system impacts significantly more users than traditional OLTP systems.

Decision support processing

Decision support system (DSS) applications provide data to enable business leaders to make informed business decisions, such as comparing sales figures between one week and another, and projecting revenue figures based on sales assumptions. The inability to execute queries can delay data analysis that is vital for key decisions.

Second-generation Web sites dynamically process information and and personalize customer experiences, which requires that DSSs integrate more closely with traditional OLTP systems with at least the same level of availability as that required by traditional OLTP systems.

For data that is loaded and unloaded into data warehouses and data marts, database availability is equally important. Organizations can load and unload data into a data warehouse as frequently as every week, and into a data mart every day. Because of the frequency of data loads, administrators often have a limited window to perform such operations. Thus, unanticipated downtime during load operations delay a user's ability to produce timely results.

Informix High Availability Features

The Informix suite of high-availability features leverages Informix's proven track record of supporting highly available databases for all types of mission- and business-critical processing. These high-availability features include online administrative utilities for backup and recovery, reorganization of tables, enterprise replication, cluster and data failover capabilities, software mirroring, and more.

High-Availability Features for Clustered Architectures

To meet the requirement for 99.999-percent availability for clustered architectures, which is higher than that which is available from a single symmetrical processing (SMP) system, Informix offers an additional set of continuous availability features that include cluster manager software, high-availability data replication, automatic client connection failover with high-availability data replication (HDR), and enterprise replication.

Cluster manager software solution

To enhance availability in multinode environments, many operating system vendors provide cluster manager solutions to connect groups of servers, or nodes. These cluster manager solutions deliver high data and application availability by providing failure detection, communication of failure to other systems and applications, system-level recovery, and restarting cluster-aware applications on the surviving node in a cluster.

The cluster manager software allows each node in the cluster to run application software independent of other nodes (Figure 1). Disks among the nodes are either shared or are easily accessible by other nodes in the cluster system, so if one node fails, the cluster manager software automatically switches the workload from the failed node to a surviving node.

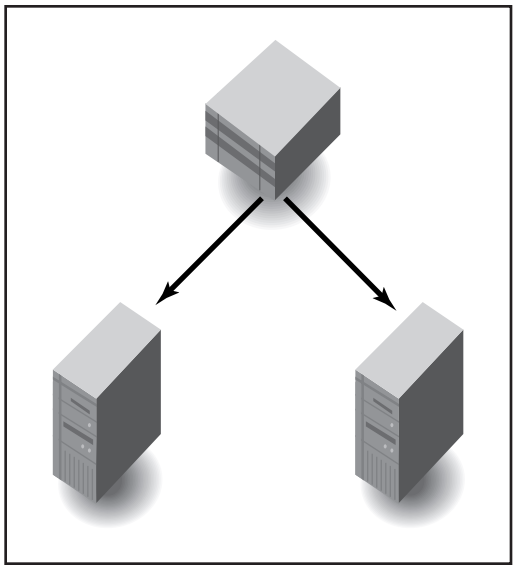

Figure 1: IDS with cluster management.

Using third party cluster manager solutions for node failure detection and notification, Informix offers a cluster failover facility to ensure continuous database processing. When a node failure is detected, the cluster failover facility automatically switches a database server from the failed node to a second node. This allows users of the failed node to continue database processing on the surviving node, significantly improving database availability between two SMP nodes.

The cluster failover process promptly creates a second instance of the database on a surviving node, and performs automatic switchover, which switches the ownership of the disks from the failed node to the surviving node. Then the Informix server performs a fast recovery operation, which restores the database to a physical and logical consistency. During this recovery process, the database is restored to the state of the last checkpoint. All of the transactions that have been omitted since the last checkpoint are rolled forward and all of the uncommitted transactions are rolled back. After the database on the surviving node has completed its recovery, the cluster software automatically restarts the applications that were running on the failed node.

The Informix cluster failover facility can be implemented in both active/passive and active/active configurations. Active/passive is a configuration in which one node runs the database server and the second node acts as a hot standby for the first node. Active/active describes a configuration where both nodes run an instance of the database server. In the event of a node failure, the surviving node acquires the workload of the failed node.

High-availability data replication

Informix Dynamic Server provides HDR, which uses two active instances of IDS. The instances can be on the same system, or two different systems. When two different systems are used, the two systems can be located anywhere because HDR replicates the primary instance to the secondary instance over a network. This replication is done by copying IDS log records from the primary system to the secondary system as they are written.

The second server is active, in a read-only mode, and operates in fast-recovery mode. The secondary instance receives the log records from the primary instance and immediately applies the log record locally. This method ensures that the secondary system is only a few seconds behind the primary at any given time. If the primary node fails, the secondary node rolls back any uncommitted transactions and then becomes the primary server. Because the secondary server is initialized and is always current relative to the primary, the secondary server can become the primary server in just a few seconds.

HDR can be configured to run in two different modes: synchronous and asynchronous. In synchronous mode transactions do not commit on the primary server until the secondary server has received each log record that comprises the transaction. In asynchronous mode each transaction is allowed to commit immediately. Synchronous mode is required for those applications that require absolute transactional consistency. Asynchronous mode is for applications that require higher performance.

Automatic client connection failover with HDR

When HDR is used in conjunction with an external cluster manager, clients are transparently reconnected to the current HDR primary. When a failure occurs, the cluster manager executes two key functions: first, the cluster manager executes scripts that convert the current secondary to the

current primary and manages the restarting of the failed server; and second, the IP addresses of the two servers are "swapped." The second action is critical because it ensures that all clients only reconnect with the current primary server. Because IP address reconfiguration is used, the change in servers is completely transparent to the client.

Enterprise replication

Data replication is increasingly used as a method to enhance database availability. Replication can minimize, or sometimes even eliminate, both planned and unplanned downtime. To ensure database availability, Informix enterprise replication (ER) replicates the entire database or a portion of the database to a secondary server (Figure 2). This option is useful for creating a hot standby server to take over processing in case the primary server fails.

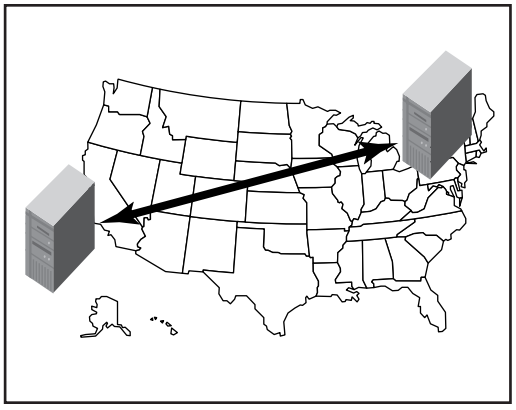

Figure 2: Geographic clusters with enterprise replication.

Enterprise replication is ideal for providing geographical availability. Because ER supports active/active replication, both sites can do useful work in normal operation. When one site fails for any reason—power, Internet cut-off, or disaster—for example, work can be switched to the other site with minimal interruption.

Informix ER also supports a full peer-to-peer replication model with update-anywhere capability. Enterprise replication protects against primary system failures by replicating data asynchronously to one or multiple secondary sites. Any updates at the primary site, including changes to the global catalog, are automatically propagated to the secondary site, ensuring that all sites have consistent replications of the data. Transmission of the updates can be immediate or time driven, in which case the database administrator (DBA) can specify the time intervals for the updates.

Updates can also be event-driven, such as after a transaction commit or as specified by the user. Informix ER employs a reliable message-delivery mechanism, which stores data locally and propagates the data to the remote server as a separate transaction. In the event of a server or network failure, the surviving server can continue to service users, providing a high degree of fault tolerance. After the failed server or network is operational, all changes to the source database are propagated to the database on the affected server.

Continuous Availability Features for SMP Architectures

To respond to the increased demand for high availability, SMP hardware vendors are building systems with greater fault resilience, incorporating components with higher mean time between failures (MTBF), stabilizing the system's operating environment, and employing technologies such as error-correcting memory, and N+1 power supplies and cooling fans. Storage subsystems are greatly improved, and technologies such as redundant arrays of inexpensive disks (RAID) and online replaceable disks and tapes can significantly increase data availability.

The database management system is another critical factor. To enhance database availability and stability on stand-alone SMP systems, Informix provides a variety of continuous-availability features, such as online maintenance and administration, fault resilience, and enhanced problem diagnostics.

Online Maintenance and Administration
To minimize database maintenance and administration downtime, Informix offers a suite of utilities that perform tasks online, such as database tuning, reorganization, backup, and recovery. For tasks that must be performed off line, data partitioning enhances availability, allowing a portion of the table to be taken off line for administration while the rest of the table remains available for user processing.

Dynamic Tuning
Regular database tuning is critical for ensuring efficient allocation of resources for fast database response time. Because the database tuning task performance often requires the database to be taken off line, administrators usually postpone these tasks until response time has deteriorated to an intolerable level.

Informix Dynamic Server is capable of performing database tuning tasks online. These tasks can be accomplished transparently, without any impact to users or applications on the system.

Database server processes can be allocated and retracted to adjust to the processing load. For example, an online retail Web site that experiences predictable surges in orders during the noon hour can dynamically add more server processes between 11 a.m. and 2 p.m. to ensure faster order processing time.

Another example of online tuning is shared-memory allocation that can be dynamically adjusted on an on-demand basis and can reconfigure memory usage online. After the memory is freed by the database, it can be reclaimed for operating system usage. Additionally, monitoring and fine tuning system parameters, such as CPU and memory utilization, asynchronous I/O queuing, available disk space, and partitioning scheme, can be performed online.

Online table reorganization

The table schema, or table reorganization, can be altered using the alter table command to add and delete a column, add, drop, and modify data constraints placed on a column, and change extent size. For example, a new column can be added in the CUSTOMER table to reflect the date of the last order to determine whether the customer is a current client, and deleting a rarely used column within a large table reduces disk space usage.

To increase database availability, IDS allows DBAs to alter table schema without rendering the table unavailable for normal use. Furthermore, alteration of a table occurs in place, so the changes are made as rows are updated without requiring a second copy of the table to be created. This improves performance and increases table availability, with minimal space requirements.

Data Partitioning

Data partitioning enables large tables and indexes to be intelligently divided into smaller partitions and distributed across multiple disks. In addition to increasing high performance and scalability, data partitioning also improves database availability. Data partitioning enables all maintenance operations, including load, index builds, backup, and recovery, to occur one partition at a time—leaving the remaining portions of the table accessible for user transactions.

An example of data partitioning is a customer order table that is partitioned by individual states. If the disk containing customer information unexpectedly fails, other partitions of the table are unaffected while the California partition is being restored to another disk. This allows users to continue to process orders for the remaining 49 states.

Informix Dynamic Server supports a wide range of partition schemes that can be monitored and tuned online when necessary:

- Simple round-robin, in which every record goes to the next partition in the sequence
- Expression, in which each partition gets a set of records based on its key values

Data Skip

Data partitioning can ensure high availability through data skip, which allows users to bypass portions of the database in the event of a disk failure. This option is especially useful during execution of a complex query, where an unexpected disk failure can force the entire query to abort. Rather than canceling the entire query, which may have taken hours to execute, the data skip option can be used to bypass the failed partition, thus allowing the query to complete its execution.

In the customer order table example, customer orders are partitioned by states. Suppose the disk containing the California partition fails during the execution of the following decision support query:

```
Select sum(total_dollar)
  where (date_year = 1997) and
        (product_code=123)
  from ORDER
```

The data-skip option can be used to skip the failed disk, allowing the query to continue summing orders for other states. After the data on the failed disk has been restored, a separate query can be issued to sum the California orders. The results can then be added to the initial query.

Alter Fragment

After a table and its associated indexes have been partitioned, they can be altered using the *alter fragment* command. Tables and indexes are partitioned and can be modified by combining tables that contain identical table structures into a single fragmented table, also called an *attached table*, or by detaching a table fragment from a fragmentation strategy and placing it into a new table, which is also called a *detached table*. Attached and detached tables are often used in situations where limited disk space necessitates moving outdated data from disk onto other forms of storage media.

Using the alter fragment command to attach and detach table fragments sometimes requires indexes to be rebuilt, which can have an adverse affect on table availability. To improve availability, the *alter fragment* command searches for reusable indexes before creating new indexes on the altered table. If portions of the existing indexes are reusable, the command instructs the server to only build indexes on the table fragment where indexes are unusable. By checking for reusable indexes, the index build process is minimized, and sometimes eliminated entirely. This results in faster response time during execution of the alter fragment command and consequently, higher data availability within the altered table. This feature is especially useful when a customer maintains a rolling window of data, such as when one table contains 12 months of data—each month within its own fragment.

A second table only contains the current month's data. At the end of each month, the table containing the current month's data can be attached as a new fragment to the 12-month table. The oldest month's data fragment is detached. This functionality provides customers with a clean, fast import of data with minimal impact to the base table.

In this example, the oldest month is June 1996. Orders are stored in dbspace db0696. The alter fragment command detaches dbspace db0696 from the ORDER table and places it into the old_ORDER table:

```
alter fragment on table ORDER
  detach
        db0696 old_ORDER
```

After old_ORDER is created, it can be copied onto another form of storage and deleted from the disk. To store orders received for the current month, a new table, new_ORDER, is created. Because ORDER and new_ORDER use the same table structure, the alter fragment command can be used to attach the two tables into a single fragmented table. The command for attaching new_ORDER to ORDER table is shown below:

```
alter fragment on table ORDER
  attach
     new_ORDER
```

When db0696 is detached from the ORDER table, if the index associated with the ORDER table uses the same fragment strategy as the table (monthly), the command instructs the server to simply drop the index fragment for db0696 and update the system catalogs. This process is much faster than rebuilding the entire index for the ORDER table. Similarly, when new_ORDER is attached to ORDER, the command recognizes that the existing index on the ORDER table can be reused. Thus, it instructs the server to only build indexes on the new_ORDER portion of the table and update the system catalogs accordingly.

Online Backup and Recovery

Database backup is an important administrative task that must be frequently performed to avoid data loss. For 24x7 operations, not only is an online backup solution critical for maintaining database availability, but an online recovery capability is equally important for bringing the database online in the shortest amount of time in the event of an unexpected failure.

Informix ON-Bar™ backup and restore utility offers various features to let administrators perform backup and restore functions without forcing the database to be brought down.

ON-Bar supports online backup, which lets administrators back up the entire database while the database continues to run. ON-Bar also supports online restore, which lets administrators recover noncritical database objects while the database server is on line. With online restore, users have continual access to the database while sections of the database are being recovered.

ON-Bar also supports dbspace-level backup and restore, allowing backup and restore operations to be performed one dbspace at a time to allow other portions of the database to remain available. In the customer order table example, the administrator could perform a backup, one state at a time, without affecting the dbspaces containing other states.

Similarly, a dbspace-level restore can significantly improve availability by enabling the administrator to restore a database to the lowest level of granularity. For example, if the database is partitioned across six disks and disk #4 unexpectedly fails, the administrator need not restore the entire database—only the dbspaces stored on disk #4. The remaining disks are available to end users for processing.

Dbspace-level restore can also enhance availability of a full system restore. If a server crash causes corruption to the database that requires the entire database to be restored from the backup media, the administrator must recover all of the critical dbspaces, such as the root dbspace, off line. After all of the critical dbspaces have been restored, the database can be brought online to recover the remaining tables. If the tables have been partitioned across multiple storage devices, they can be restored in parallel to speed the recovery process. After a dbspace has been restored to a physical and logical consistency, the data in that partition can immediately be made available to users for transaction processing while the remaining dbspaces are being recovered.

Informix administrators also have the option to use incremental backup. Using the date/time stamp located on every database page, incremental backup copies only those pages that have been modified, rather than backing up an entire partition or table. With large tables that have relatively few updates, incremental backup can significantly reduce the time for performing backup operations—ensuring data protection while providing the highest level of database availability.

External Backup

The ON-Bar backup and restore utility provides an effective means for creating an internal backup of the database that can be used to ensure database availability in the event of failure. To further assure availability, customers usually create an external system backup for disaster recovery. External backup uses proprietary hardware and software technologies to create simultaneous copies of data to host independent, local, and remote sites. In this way, external backup is faster and can be restored in a similar setup environment.

To enable external backup, Informix servers provide an administration command to force a checkpoint, which flushes the buffers to the disk and blocks the server from accepting any implicit or explicit transactions. After the external backup has been performed, another command is issued to undo the blocking, and normal server operations can be resumed. Users can then perform an internal backup using the ON-Bar backup and restore utility.

Restartable Restore

Sometimes an I/O error occurs on the tape, or other errors within the servers can occur during a physical or logical restore. When this happens, the entire restore process must be restarted from the beginning. To decrease the time required to perform a restore following an error, restartable restore allows the restoration to be restarted close to where the original restore failed. Depending upon how much data must be restored and where the data error occurred, this feature can significantly improve server availability.

If a user performs level-0, level-1, and level-2 backups for dbspace1, dbspace2, and dbspace3, and a restore of the three dbspaces is attempted and fails during the level 1 restore of dbspace2, the restarted restore performs the level 1 and level 2 restores for

dbspace2, and level 0, 1, and 2 restores for dbspace3. The restore for dbspace1, as well as the level 0 restore for dbspace2, are skipped because they were successful during the original restore. During the logical restore portion, this feature enables the server to replay logs, starting with the log that had the most recent checkpoint before the error occurred.

If a failure occurrs during a physical restore, restartable restore restarts the restore at the last non-complete dbspace. If a failure occurs during a logical restore, restartable restore restarts from the last checkpoint. Restartable restore is supported in both cold (off line) and warm (online) recovery of a physical restore. However, restartable restore for logical restore is supported only during cold recovery.

Oncheck Utility

Informix provides a complete suite of utilities to ensure full data integrity and optimal data consistency. The Oncheck utility performs checks to search disk structures for inconsistencies, repairs index structures that contain inconsistencies, and displays information about the disk structure.

To increase table availability and improve concurrency, Oncheck eliminates the need to lock the table while checking indexes. This allows users to continue to access the database while checks are being performed. By eliminating the requirement of placing locks on a table while it is being checked, Oncheck significantly enhances concurrency while ensuring an optimal level of consistency.

Oncheck no longer requires that physical and logical logs be checked during reserved page checks because the reserved page check needs to be fast so that a server that is down can be brought back on line quickly. For this reason, Oncheck lets the user decide whether or not the logs should be checked during the operation.

Fault Resiliency

Informix servers offer a host of features that are designed to work around any faults that may cause a database to shut down. These features include database and log mirroring, fast recovery, enterprise replication, and cluster failover.

Database and Log Mirroring

Database and log mirroring provide database administrators with a means of recovering data in the event of a media failure, without having to take the database server off line. This method is ideal for protecting critical data that requires high reliability. Examples of data that should be mirrored include root dbspace, and logical and physical log files. If the media that stores any of these data fails, the database is immediately taken off line.

By supporting database and log mirroring, Informix servers give the administrators the option of mirroring only the portion of the database that requires high availability. For example, if disk 1 contains two tables: CUSTOMER and RECEIVABLES, the administrator may consider CUSTOMER to be a critical table, and RECEIVABLES to be less important. With database mirroring, the administrator can choose to only mirror the CUSTOMER table.

Informix Dynamic Server supports hardware and software mirroring when provided by the operating system, system software, and underlying hardware. Unlike database mirroring, where mirroring is achieved at the database level, hardware mirroring is achieved at the disk level. Consequently, the entire disk is mirrored, eliminating the flexibility to select which portion of the database to mirror. Therefore, with the customer order table example, hardware mirroring forces the administrator to mirror both tables on disk 1, which can be a tremendous waste of disk space.

Fast Recovery
Unexpected shutdowns can occur, despite preventative measures. Fast recovery is an Informix server utility that brings the system online quickly and without data loss to maintain full data integrity.

When invoked during a system recovery from an abnormal shutdown, fast recovery applies the transaction logs to the data files to restore the database to a state of physical and logical consistency. During this recovery process, the database is restored to its state at the last checkpoint. All of the committed transactions since the last checkpoint are then rolled forward and all of the uncommitted transactions are rolled back.

Exception Handling
A failure within a session often causes an entire server to shut down with an assertion failure. These server failures can be prevented by isolating the errors at the session level so they do not affect the remaining server processing.

Informix servers provide a set of routines to handle assertion failures and warnings within the server. These routines minimize server downtime by effectively pinpointing and diagnosing the problem areas, and returning appropriate error messages indicating what has transpired. For unavoidable server failures, these exception-handling routines provide better diagnostic information to assist in finding and fixing the problems.

Enhanced Problem Diagnostics
In the event of a server failure, Informix servers offer several enhancements to assist Informix technical support with problem diagnostics, analysis, and resolution. These enhancements help pinpoint the problem areas more quickly, thereby allowing users to bring the server back on line as quickly as possible.

Smarter diagnostics consist of enhancements in six areas: event alarms, fault isolation, shared memory dumps, stack tracing, additional utility options, and thread blocking routines. These features provide quick resolution of reported problems.

Conclusion

To respond to the increasing demand for higher database availability, Informix offers a wide range of features to provide around-the-clock database processing. These features minimize planned downtime by allowing administrators to perform database maintenance operations online and they reduce the impact of unplanned downtime by working around any faults that may occur. Combined with high-availability features provided by hardware vendors, Informix ensures a continuous database processing environment ideal for mission- and business-critical processing.

About Informix

Informix Software, the database company, is a leading provider of database management systems for data warehousing, transaction processing and eBusiness applications. With more than 100,000 customers worldwide, Informix Software delivers high-performance database systems in markets including retail, financial services, government, health care, manufacturing, media and publishing, and telecommunications. For more information, visit the Informix Web site at www.informix.com.

Informix®
SOFTWARE
The database company™

4100 Bohannon Drive
Menlo Park, CA 94025
Tel. 650.926.6300
www.informix.com

INFORMIX REGIONAL SALES OFFICES

Asia Pacific	65 298 1716	Japan	81 3 5562 4500
Canada (Toronto)	416 730 9009	Latin America	305 591 9592
Europe/Middle East/Africa	44 208 818 1000	North America	800 331 1763
Federal	703 847 2900		650 926 6300

Printed in U.S.A. 5/01
000-22314-70