White paper

Cisco and IBM Informix Database Software: Enhancing Collaboration

Highlights

This white paper addresses a number of the features and techniques Cisco uses to embed Informix in its solutions, and why .

Contents

Introduction

Cisco® Collaboration solutions help people work together to solve problems, make decisions, accelerate time to market, and transform entire industries faster, no matter how far apart the participants happen to be located. Businesses of all sizes use Cisco Collaboration solutions, from small businesses to large enterprises. In fact, IBM® has a sizeable deployment of Cisco Collaboration technology across its offices worldwide.

The infrastructure required to power collaboration solutions like this needs to be robust, scalable, and invisible. Cisco chose to embed IBM Informix® database software in their solutions because of the robust embeddability, high reliability, and extreme performance of Informix. Today, Cisco has over 100,000 instances of Informix running in production environments; a number that continues to grow. When a customer makes a VoIP call using a Cisco Collaboration solution, they have unknowingly just used Informix! The number of product teams inside of Cisco using Informix is also growing. As a result, Cisco developed its own internal competency center around Informix so that its product groups are leveraging best practices and have an internal support community, in addition to the enablement support provided by IBM.

This white paper addresses a number of the features and techniques Cisco uses to embed Informix in its solutions, and why. It also highlights some of Informix's newer features. The white paper is divided into six sections, each corresponding to the different tasks involved in the embedding process.

Deployment

Deployment is the first step in embedding the database in the target application or device. It establishes the delivery of software onto the platform on which it will be utilized. An embeddable database deployment provides an organized, repeatable mechanism that allows for easy and customizable integration with the host application or device. Its scope covers packaging and deployment methodologies, footprint optimization, software configuration and upgrades. Since user interaction in an embedded environment is designed to be non-existent or minimal, the installation process should not expect user input. The deployment of the database software must itself be embedded within the installation application.

Using the Installation Application to Install the Database

Cisco utilizes the installer of its collaboration applications to configure and deploy the Informix database server. It is built on a Java-based framework and allows for a customized installation. To use this approach, Informix is installed using an unattended installation. An installation template is first created by running the Informix installer in a template environment with the option to generate a response file. The template represents a snapshot of the desired deployment configuration and consists of entries in a response file which can be customized. Alternatively, a copy of an existing or sample response file can be modified to fit the deployment preferences on the target system.

To deploy Informix, the installation application is started on the target system and provided with the response file containing the custom installation configurations as input, along with the Informix installation binaries.

For Cisco, the ability to use automated deployment provides a way to perform a mass deployment of Informix onto all the devices and in-house systems needed, which in large enterprises can be numbered in the thousands.

Using the Informix deployment Utility – An Alternative

A second and newer deployment alternative uses a lightweight, space-conscious approach for deployment. It uses the Informix deployment utility which ships as part of the Informix installation ($INFORMIXDIR/bin/ifxdeploy). The deployment utility facilitates the deployment of a pre-configured Informix installation snapshot and server instance. An interactive installation of Informix is performed and the database configured the way it would be in a production environment. Once configured, a snapshot is taken for deployment.

Cisco chose to embed IBM Informix® database software in their solutions because of the robust embeddability, high reliability, and extreme performance of Informix.

In contrast to the installation application approach, this alternative requires the explicit archiving of a configured instance. To manually package an instance, all open transactions must be closed to maintain data integrity. The installation directory, configuration settings (server configuration file, connectivity configuration file), and data (optional) are then archived using an archiving mechanism of choice. Finally, the target must have the capability to extract the archive.

The deployment assistant utility ($INFORMIXDIR/bin/ifxdeployassist) is used to archive a snapshot of an Informix instance. This utility provides both a graphical interface and a command-line interface (CLI) through which an Informix instance can be intuitively packaged. Packaging preferences can be customized by specifying which Informix features should be packaged, which server instance should be packaged, what data to include, and so forth.

Cisco is currently exploring the deployment utility and deployment assistant utility for use in its next generation of offerings.

Cisco® Collaboration solutions help people work together to solve problems, make decisions, accelerate time to market, and transform entire industries faster, no matter how far apart the participants happen to be located.

Upgrades

When a new version of a product is released, it is common for a customer to upgrade to take full advantage of new features and functionality. Depending on the type of solution being used, an upgrade could involve multiple aspects including the hardware, operating system, database software, other middleware, and the application software itself.

When the database software is upgraded, it is critical that data integrity be preserved. In an environment containing multiple Informix installations and instances, the following upgrade sequence of steps is repeatable to facilitate mass upgrades.

Upgrade Steps

To maintain data integrity and increase the success of an Informix server upgrade, the following steps are performed prior to deploying the newer server version:

- Close all open transactions
- (Optional) Back up the data. Informix provides several options to accomplish this such as the ontape and onbar utilities, or alternatively using the Conversion Guard utility
- Shut down the database server
- Begin server deployment for upgrade

Informix supports a number of upgrade methods, each providing unique capabilities. The different methods are summarized in Table 1 below:

Table 1. Methods

Method Name	Description
Overwrite	The target database server installation binaries overwrite the existing source database server binaries.
Preserve	The target database server is installed in a different installation location from the source server and applications are pointed at the new location.
ifxdeploy utility	This deployment utility ($INFORMIXDIR/bin/ifxdeploy) is used to perform upgrade deployments.

Once the deployment of the new version is complete, the server environment is established by setting the required and preferred environment variables to reflect the new server settings. When the new server is initialized, the server goes through a conversion process which updates the internal server structures to the latest release. Features that have changed in the new release are also upgraded by this process.

Some customers tend to upgrade the entire stack (hardware and software) at once, versus individual pieces. Therefore, from a database perspective, the need to export existing data and migrate it to the new server is critical. Informix offers a number of migration tools to assist in such a process. The *dbexport* utility exports the schema and data in a database-independent format which can then be re-imported with the *dbimport* utility into a new Informix database. Another method of migrating data is by taking a backup on the existing server and restoring it on the target server. Cisco is exploring the use of the High Performance Loader in Informix to quickly unload an existing database and reload it on the new server. This is a highly scriptable process, and internal tests have demonstrated it can unload about 75 GB per CPU per hour. On a four-CPU machine, this results in an unload rate of 300 GB per hour.

Footprint Optimization

When a solution is installed in the end customer's environment, the installation process must be able to determine the nature of the hardware it is installed on and configure itself accordingly. For example, a call center that handles 5000 calls a day may choose to install the solution on a smaller server configuration than the call center that handles 5,000,000 calls a day. The subject of footprint optimization is at the heart of most embedding requirements. Embedding implies that the application will exist within the context of the host, that is, the software or hardware within which it is embedded. The host also has to exist within the constraints imposed by the physical device, including footprint limits.

Informix provides several capabilities to facilitate footprint management. In the context of deployment, footprint optimization consists of the amount of space the application consumes on disk, as well as the amount of system resources it consumes when operating. The size of a full Informix database server installation that includes most features and multi-lingual support is typically less than 250 MB (excluding data). In an embedded environment with physical resource constraints, 250 MB may be too high. Furthermore, certain Informix features or utilities may not be used. Therefore, a requirements gathering phase is an essential first step in the footprint optimization process.

The Informix installation footprint size can be controlled during template instance creation. The administrator would select the custom installation type and select the list of desired features. The deployment assistant provides further capability to customize the footprint during snapshot creation. It detects the installed features and presents a feature tree that allows the administrator to de-select features that are not needed while creating the snapshot. This extended flexibility allows the administrator to customize each snapshot without having to install Informix several times. Furthermore, the deployment assistant allows the administrator to specify whether or not to include data in the snapshot. It may be desirable to deploy an instance with data to reduce the initialization time.

In addition to configuring the disk space used by Informix, memory and CPU utilization can also be tuned. The configuration of system resources in this manner allows Informix to run within acceptable performance limits of the machine. In the background, Informix is made up of different virtual process and thread classes, each performing specialized functions. Before the database server is initialized, configuration parameters (e.g. MULTIPROCESSOR, VPCLASS, AUTO_AIOVPS, SINGLE_CPU_VP, and NETTYPE) can be set to increase the utilization of the different virtual classes.

Just as the Informix CPU configuration parameters impact the utilization and load on physical processors on the system, the Informix shared memory configuration impacts the amount of physical and virtual memory allocation made on the machine. The typical Informix database server installation uses three shared memory sections: resident, virtual, and message.

The resident memory caches user data in memory for faster access and also tracks resources used by the database. It contains the bufferpool, the physical log buffers, logical log buffers, and least recently used (LRU) queues. The virtual memory is used for resource management for database processes. These include thread management information, user-session data, database dictionary information, stored procedures, etc. The message memory is responsible for communication management and coordination between database server and application processes. It contains message buffers for local shared memory communication to and from the server. There are several Informix configuration parameters to tune memory usage. These parameters affect behaviors like the amount of data and information buffered, the length of time for which they are buffered, and so on. The parameter used to set the maximum shared memory usable by the database server is the SHMTOTAL configuration parameter. It should be set appropriately for each environment.

Cisco uses Informix to store configuration, user, and reporting data. For configuration data, the runtime footprint of the database is typically small, and the memory and disk footprint is reduced at installation time. For reporting applications, the runtime footprint is much larger and consumes a significant portion of the server resources. While the disk footprint is constrained to a pre-determined size and data is pruned to maximize available space, memory is automatically increased as needed by the engine up to a maximum predefined value.

Backup and Restore

The ability to make a copy of a database for backup purposes or to move the copy to another location is crucial to ensuring the recovery of data in the event of a system failure. It is important to be able perform this task without taking the database offline so that system outages can be avoided. Since an active database is continually modifying disk pages across whatever disks or disk files it is using, special backup utilities are traditionally used because manual file copying methods will not work. Informix offers several backup strategies that can handle both small and enterprise-size databases.

The Informix *ontape* utility is a simple backup utility that can create a consistent point in time image of an active Informix database. This backup image can be written to a file, a tape, or to a pipe, where it can be optionally compressed, then sent to a destination for restoring. *Ontape* supports full, incremental, and delta backups. It can also back up transaction logs as they are completed or in batch.

Cisco makes wide use of the *ontape* utility in conjunction with a *compression* utility to provide a fast and reliable backup mechanism with a small footprint. The *ontape* utility is further used to quickly image secondary nodes in a database replication environment. The implementation of backup and restore processes are scripted so that the customer can execute them either with the click of a mouse or schedule them to run automatically through a control interface.

System Monitoring Interface

In an embedded environment where there is no user interaction, it is critical for the integrated application to be able to detect, diagnose, and automatically fix any problems that arise. Diagnosing problems with the database server has become easier and faster with newer versions of Informix. Informix provides both reactive and proactive methods for diagnosing issues with shared memory, transaction log files, disk space administration, and more. Some of the available methods for diagnosing problems include event alarms, the message log, the *oncheck* and *onstat* utilities, the *System Monitoring Interface* (SMI) tables, and the system console.

Informix provides an extensive event alarm system for automatically triggering administrative actions based on internal events that occur. The application can define its own alarm handling program and decide upon the appropriate corrective action for each of the event alarm classes. A typical example would be the automatic setup of database replication.

The SMI tables—special tables that contain dynamic information about the state of the database server—can be queried from the integrated application using standard SQL queries. These queries can be combined and used with the Database Scheduler to provide an automated and proactive health-check mechanism directly inside the database server. In addition, Informix has built-in capabilities to look at the server configuration and transaction workload to determine if it has been tuned optimally. If it detects any discrepancies, it generates performance advisories, which include recommended changes to improve performance, and saves them in the message log for future action.

Informix has built-in capabilities that allow it to automatically respond to common database conditions and manage its resources for optimum performance. For example, non-blocking checkpoints can be configured automatically, so that Informix adjusts its memory page write frequency (LRU flushing) when the workload becomes performance-intensive to prevent transactional blocking activity during a checkpoint. Informix has the ability to automatically trigger checkpoints to help maintain optimal performance. With the use of pre-allocated files (cooked chunks) for storing data becoming more prevalent, Informix can automatically monitor the I/O performance and tune the number of I/O threads (AIO VPs) accordingly. Many database configuration parameters have an automatic setting, which allows Informix to determine their optimal values over the course of time.

Informix provides programmatic access to most of the administrative commands through the SQL Administrative API. This includes typical administration tasks such as adding space, adding or dropping a logical log file, and so on. Advanced administrative actions for managing replication are also possible using the administration API. This allows database administration to be fully-automated, eliminating the need for end-user or DBA interaction.

Storage Space Management

It is critical that an embedded database server monitor and manage disk space to avoid out-of-space conditions. Without a mechanism to track the used space and allocate more space as needed, the database would ultimately fail when the allocated space became exhausted. This may not only result in an expensive outage, but may also compromise the integrity of a system capturing sensitive data.

Informix supports two interfaces for storage space management: a CLI and an SQL interface. The command line utility, *onspaces*, is a space management utility used to create data storage spaces, add additional space to existing storage spaces, initiate mirroring for storage spaces for data redundancy, and delete storage space. Using the *onspaces* CLI allows for easy space management scripting. Depending on deployment requirements, *onspaces* can make the necessary call(s) to administer storage spaces.

In some embedded scenarios, the total available space is pre-defined and the application must manage the use of that space. In the Cisco collaboration applications, disk space is fixed when deployed, requiring the use of *onspaces* utility scripts only as the preferred mechanism to define and allocate database storage.

Other embedded scenarios require capabilities to handle variable storage requirements, such as when the disk characteristics of the system on which the application will be deployed are usually not known beforehand. The new automatic storage features in the Informix 11.7 release allow the database server to be configured for automatic storage space expansion using a pool of storage allocated to Informix. The two mechanisms through which storage space can be automatically expanded are adding new chunks or extending existing chunks.

Cisco uses a partitioning scheme to manage several hundred gigabytes of stored data. On a daily basis, a new partition is added to each of the major tables to accept data for that day. When the data gets too old or there is need for additional space, the oldest of these partitions is dropped for each of the tables. This helps to ensure that related data is aged off at the same time. This also prevents fragmentation of data within the table over time as would occur with a regular SQL delete/insert strategy. This operation takes on the order of one second for the entire database, regardless of whether there is one MB of space to clear and re-add or 10 GB.

High Availability and Replication

In a company that must be available all the time, data must also be always available. IBM Informix provides a wide variety of high-availability and clustering solutions that maintain the availability of data. A proven and widely-used solution which has been around since 1994 is the High Availability Data Replication (HDR) feature, in which a secondary server maintains a backup copy of the entire primary server through synchronous or asynchronous data replication. Applications can access the secondary server quickly if the primary server fails.

Another frequently used technology is Enterprise Replication. Enterprise Replication implements asynchronous data replication at the table or database granularity between two or more Informix database servers. In this solution, network latency and target server outages are tolerated. In the event of a database server or network failure, the local database server continues to service local users while accumulating transactions in persistent storage until the target server becomes available. Enterprise Replication ensures that data reaches the appropriate server efficiently with a minimum amount of copying and sending data.

Cisco is an innovator in designing Enterprise Replication-enabled, embedded solutions for the enterprise market. Cisco delivers clusters of over a dozen instances with write-anywhere capability. This follows Cisco's scaling plan of increasing capacity through the addition of low-cost components (as opposed to scaling up a single server) and allows customers to scale vertically as well as geographically, providing fast local access to a single global data image. Cisco uses a number of Enterprise Replication alarms to determine the Enterprise Replication state, so that the application can monitor the replication and fix any problems. For example, the application can make use of an event alarm class to handle unexpected circumstances such as low memory. The addition of new nodes to the cluster is completely automated, as well as the recovery from connectivity loss or downed systems. The result is a highly reliable, available, automated system whose internal operations are hidden from the end user. From the user's perspective, it is a system that just works.

Conclusion

There are a number of technological differentiators that make Informix stand out from the competition for embedded data management. This white paper focused on the features that Cisco uses; however there are others that make Informix a great choice for other embedded scenarios as well. In addition to its proven reliability, performance, and minimal maintenance, Informix helps enable Cisco to get to market faster with complete world-class collaboration solutions that shield end users from data management tasks and provide a high-quality, reliable, and positive experience for Cisco customers. Informix offers flexibility and innovation in the crucial embeddability areas of deployment, upgrades, footprint optimization, space management, monitoring, self-management, and high availability. With Informix powering Cisco solutions, customers can be confident that their enterprise-class collaboration solution is built on a solid foundation. This is additionally backed up by a strong partnership and close collaboration between the Cisco and IBM research and development teams.

For more Information

For more information on Cisco Collaboration Solutions please visit: www.cisco.com/go/collaboration

For more information on IBM Informix database software visit: www.ibm.com/software/data/informix/embed

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
www.cisco.com/go/ibm

International Business Machines Corporation
New Orchard Road
Armonk, New York 10504
www.ibm.com/cisco

THE INFORMATION HEREIN IS PROVIDED "AS IS," WITHOUT ANY WARRANTIES OR REPRESENTATIONS, EXPRESS, IMPLIED OR STATUTORY, INCLUDING WITHOUT LIMITATION, WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.