

Technical Comparison of Oracle Database vs. IBM DB2: Focus on Information Integration

*An Oracle White Paper
January 2003*

Technical Comparison of Oracle Database vs. IBM DB2: Focus on Information Integration

Executive Overview	4
Introduction	4
Key Differentiators	6
Detailed Comparisons for Data Replication.....	8
Installation and Configuration.....	9
Key Features	9
Capture	10
Staging and Propagation.....	12
Rules and Subsetting	13
Data Transformation and Apply.....	14
Conflict Detection and Resolution	15
Configurations	16
Manageability.....	16
Detailed Comparisons for Message Queuing	17
Installation and Configuration.....	18
Key Features	18
Integration Model	18
Message Payload type	19
Publish/Subscribe and Point-to-multipoint Communications.....	20
Message Gateway.....	21
Message Queuing Functionality.....	21
Manageability	22
System Management	23
Message Management	23
Performance	24
Detailed Comparisons for Synchronous Data Access (Federated Database).....	25
Installation and Configuration.....	28
Data Sources Supported	29
Platform Availability.....	29
Key Features	30
Distributed SQL	30
Heterogeneous Distributed SQL	30
Location Transparency	31

Distributed Update Support.....	31
Distributed Transaction Support	32
Optimization of Distributed Queries	32
Materialized Views	33
Performance	33
Conclusion.....	33

Technical Comparison of Oracle Database vs. IBM DB2: Focus on Information Integration

EXECUTIVE OVERVIEW

This paper provides a technical comparison of the information integration features of Oracle9i and DB2 v8.1. Oracle9i offers a full suite of features for the distributed environment that is vastly superior to IBM's offerings.

Oracle offers a single, unified solution that satisfies the complete spectrum of information sharing requirements, Oracle9i. It enables the coordination of disparate processes, applications, and systems so that they can be centrally controlled and can function together as a unified whole. Unlike IBM's offering, which requires a separate product for every feature, Oracle offers all the features as part of one product, the Oracle database.

Oracle9i is the only complete solution for operating in a distributed environment on the market today. This paper will clearly establish Oracle9i as the leader and leave no doubt that DB2 is no competition for Oracle9i in the distributed space.

INTRODUCTION

Oracle offers a single, unified solution to satisfy the complete spectrum of information integration requirements, Oracle9i.

Oracle9i is the leading database in the industry that allows organizations to efficiently store, manage, integrate and use business information. It is a complete solution. Oracle9i has the ability to integrate and consolidate all information for an organization from a central point. By enabling an integrated and complete solution for all business information, Oracle9i, maximizes the efficiency and minimizes the cost of information sharing, thus allowing organizations to easily and quickly take advantage of the synergies inherent in business information.

Oracle9i offers several solutions for sharing information, within, as well as, outside an enterprise. There are two scenarios for information integration: synchronous access and asynchronous access. Synchronous access uses Oracle9i Distributed SQL features to consolidate information on the fly, masking the location of objects from the application or user by making objects in the remote data source, appear local (IBM calls this federated database). Asynchronous integration uses messaging and replication technologies to move data from a

remote database to a local database, where applications can directly access the data.

Oracle9i Release 2 introduces a new asynchronous information integration feature, called Oracle9i Streams. Oracle9i Streams enables the propagation of data, transactions and events in a single data stream or queue, either within a database, or from one database to another. Information in the stream is routed to subscribed destinations. The result is a new feature that provides greater functionality and flexibility than traditional solutions for capturing and managing events, and sharing the events with other databases and applications.

Oracle offers several technologies for integrating information in a heterogeneous environment. These features extend Oracle's capabilities to work with non-Oracle data sources, non-Oracle message queuing systems, and non-SQL applications, ensuring interoperability with other vendor's products and technologies.

IBM on the other hand takes a very different approach. Instead of the database being the center for information integration, IBM offers separate products for each integration scenario.

Let us take a look at an integration scenario using IBM's products. If a customer wants to have multiple copies of the data they would have to install and configure a product called DataPropagator. If that customer wanted to asynchronously exchange information among different applications, then they would have to purchase another product, WebSphere MQ. Now if this customer wanted to integrate information from data sources other than DB2, well then they would have to get yet another product, DataJoiner. This example makes it clear that one product is not enough. Several products are required to have information integration using IBM's approach.

With IBM's offerings the customer is investing a lot of time installing all the separate products, configuring each one and then getting them all to work together. Just getting all the products to coexist is a huge burden in itself and we have not even gotten to the part of actually integrating the business information.

With Oracle's offering there are no such roadblocks. Oracle has one product that is the engine and center for information integration. It is the Oracle database. Oracle9i enables the coordination of disparate processes, applications, and systems so that they can be centrally controlled and can function together as a unified whole. Customers can replicate data, exchange information internally or externally, process and route the information based on predefined rules and integrate information from non-Oracle sources using a single product, Oracle9i. The benefit is one product, one install, and one single unified solution.

There is another key advantage of having a single product. Since all the information integration solutions are part of the Oracle database, they automatically inherit the security, reliability, high availability and recovery

features of Oracle9i. Information integration applications can use the transparent fail-over capabilities of real-application clusters (RAC). As the database adds new features in these areas they are all immediately available to the information integration applications.

With IBM's products, customers have to deal with different security and reliability models – one for the DB2 operations and the other for WebSphere MQ. They have to recover DB2 and they have to recover WebSphere MQ. There is no transparent fail-over functionality for DB2 and for WebSphere MQ.

Key Differentiators

The table below summarizes the key differentiators between Oracle and IBM in the area of integration.

This paper will describe key differentiators in each of the following areas of information integration:

- Data replication
- Message queuing
- Synchronous data access

Feature	Oracle	IBM
Replication		
Integration with Database	Oracle Streams is an integrated feature that functions identically on all hardware platforms	Replication functionality and behavior differ depending on hardware platform
Flexible replication topologies	Independent capture, propagation, and apply capabilities of DML and DDL changes across multiple databases support any topology	Point-to-point capture and apply of table data only—Unmanageable with large numbers of databases
Update anywhere support	Transformations enable update anywhere replication between non-identical database objects	Update anywhere requires identical copies of objects at each database
Replication of DDL	DDL executed and replicated on the fly	DDL not replicated

Feature	Oracle	IBM
Message Queuing		
Single solution for both message queuing and SQL operations	Message queuing is an integral Oracle database feature	Different database and message queuing products.
Single recovery, high availability, and scalability solution for SQL and message queuing operations	Database recovery recovers both SQL and message queuing operations. Scalability and high availability of Oracle database applies to both message queuing and SQL operations	Recovery, high availability, and scalability have to be configured differently for message queuing and SQL operations. Presence of distributed two-phase commit adversely impacts the scalability of combined SQL and message queuing operations.
Integrated database and message queuing operations	Single transaction, security, and data model for SQL and message queuing operations	Requires distributed two-phase commit for transaction coordination Different security and transaction model for SQL and message queuing operations
Feature	Oracle	IBM
Synchronous Data Access		
Data sources supported	DB2, Sybase, Microsoft SQL Server, Teradata, Ingres, Rdb, RMS, Informix	Oracle, Sybase, Microsoft SQL Server, Informix
Distributed update support	Oracle supports update of the non-Oracle data source	IBM only supports read access to non-IBM databases
Distributed Transaction support	Oracle supports heterogeneous distributed transactions	IBM has no support for heterogeneous distributed transactions

Table 1: key differentiators

DETAILED COMPARISONS FOR DATA REPLICATION

Oracle has been providing replication technology since Oracle7. This technology is used in many diverse configurations throughout the world. Oracle Advanced Replication maintains identical copies of data at n-way connected databases in near-real time, and Oracle Materialized Views maintains point-in-time copies of data, typically used in mass deployment environments. The guiding principle of Oracle's replication features is to ensure data integrity while allowing data to be updated anywhere.

In Oracle9i, this fundamental principle is extended with Oracle Streams. Oracle Streams facilitates the management and propagation of data, transactions, and events in a data stream either within a single database or between multiple databases. This feature provides functionality and flexibility in information sharing far surpassing most traditional replication solutions. Not only can Oracle Streams capture, propagate, and apply changes to data, it can also handle data structure changes (DDL) and user-defined events. Changes can be captured and applied as is, or transformed at any point in the capture, propagation, and apply processing. Utilizing a set of rules for each process, the customer exercises full control over the actions of Oracle Streams.

A copy of a replicated object does not have to be identical to its source object. For example, it can have different column names, or only a subset of the columns or rows. The target database for the replicated object does not even need direct connectivity to the source site. Oracle Streams provides a routing ability where changes can be propagated through multiple intermediate database staging areas before reaching the final apply site. This directed network capability exists independent of the apply process at any intermediate site; that is, the apply process does not have to apply the change at the intermediate site before routing the change to its next destination. This feature is particularly useful in wide area network environments, where the bandwidth between all sites is not equal.

Finally, Oracle Streams provides a unified environment for information sharing that supports explicit enqueueing and dequeuing of events, in addition to default data capture and apply mechanisms. This allows developers to build applications that leverage the features of both replication and message queuing, enabling the creation of an entirely new class of applications.

IBM's replication technology, on the other hand, is very limited compared to Oracle Streams. Architecturally, it is similar to Oracle's materialized view technology. IBM's technology, however, is primarily read-only. Although limited support for bi-directional replication is supported, the replicas must be exact copies of the primary. Unlike Oracle's technology, IBM does not support subsetting of updateable replicas.

IBM does not provide a generalized information sharing architecture that integrates message queuing and replication technologies. Rather developers

must choose between using message queuing and replication features, and cannot combine the two.

Unlike Oracle's replication technology, which is fully integrated with the Oracle database, some IBM platforms require the use of add on technology, most typically DB2 DataPropagator, to support replication. There are different versions of DataPropagator for different platforms (for example, DataPropagator for Z/OS, DataPropagator for iSeries, and so on). Replication does not function identically across all platforms. With Oracle, all replication features function identically on each platform.

The following sections compare the functionality of Oracle's Streams-based replication versus IBM's replication in the following categories:

- Installation and configuration
- Key Features
- Manageability

Installation and Configuration

Oracle Streams is a feature of the Oracle database. There are no additional components to install. When the database installation is complete, Oracle Streams is immediately available for configuration and use.

Oracle Streams supports heterogeneous replication via a gateway. The Oracle Transparent Gateways are licensed separately from the Oracle Server and are available on a variety of platforms, depending on the gateway.

Oracle Enterprise Manager, bundled as part of the Oracle Server packaging, provides a graphical user interface for configuring a replication environment, available on most platforms. Oracle Enterprise Manager provides both configuration and monitoring capabilities for all Oracle replication features. In addition to automatically configuring the environment, Enterprise Manager can be used as an educational tool to generate scripts demonstrating usage of the streams PL/SQL API provided.

To support replication between DB2 databases, IBM customers running on Z/OS and OS/390, or iSeries must install the appropriate version of DataPropagator. This product is sold separately from the database. Heterogeneous replication requires support for federated access to the non-IBM database, which will not be provided until a future release of DB2.

Key Features

Similar to Oracle Streams, DB2 replication consists of a capture phase and an apply phase. Once the data is captured and placed in a staging area, both products support either local or remote apply. Oracle further supports

propagation between staging areas, allowing for complex routing of data. Both products support bi-directional replication with conflict detection. Whereas Oracle Streams provides a variety of conflict resolution strategies on replicas that can contain a subset of the replicated data, IBM's conflict resolution strategy is limited to discard model and all updateable replicas must be identical to the primary (parent) site. These features of replication are examined in more detail in the following sections.

Capture

Oracle	IBM
DML and/or DDL changes can be captured	DML changes only ¹
Captured changes are stored in one (or more) high-performance queues	A separate change table is maintained for each replication source table ²
Piecewise updates to LOBs are captured and applied.	Entire LOB is read from source table. No ability to replicate just the updated pieces of the LOB ³ .
Applications can explicitly enqueue change records using standard APIs (JMS, SOAP, C, PLSQL)	No standard APIs supported for applications to insert change records

Table 2: Capture Summary

Both Oracle Streams and IBM's Data Propagator products use a log-based capture mechanism; however, the Streams mechanism offers considerably more flexibility.

Oracle Streams can capture both DML as well as DDL changes. Streams uses hot mining to capture changes as they are committed to the active redo logs. Captured changes are reformatted into logical change records (LCR) and enqueued into a queue table at the source database based on user-specified rules that govern the data to enqueue. A single capture process can capture changes to multiple tables across multiple schemas or even the entire database, both DML and DDL.

Oracle Streams seamlessly handles changes in data structures by maintaining the source catalog versioning information at the target sites. This information is captured automatically whenever a structure change occurs.

¹ DB2 Replication Guide and Reference, Version 8, page 185.

² DB2 Replication Guide and Reference, Version 8, page 38.

³ DB2 Replication Guide and Reference, Version 8, page 99.

In addition to the default capture process, users can elect to enqueue their own events or LCRs. The capture process can be parallelized so that changes can be reformatted and enqueued in parallel. The capture process is tunable with process parameters that can be set while capture is running. Dynamic parameter changes are persistent across capture and database shutdown. Finally, multiple capture processes can be run concurrently.

IBM captures all DML changes to a replicated table that is registered, although some subsetting is supported for read-only replicas. IBM recently added support for adding a column to a replicated table, but it still requires special processing to actually replicate this column⁴.

The IBM capture program captures changes that are made to the source. The IBM capture program adds a row(s) to the change data (CD) table for each change that it finds in the log. Each replication source has a CD table. Information about committed transactions is stored in the unit-of-work (UOW) table. For updatable replication, during the apply phase, the apply program joins the CD and UOW tables to ensure that only committed changes are copied over to the target⁵. These tables are typically located in multiple databases, requiring expensive joins across the network in order to apply changes.

LOBs are never written to the CD table. The entire LOB is read by the APPLY program from the source table directly and applied across the network. This can cause data inconsistency issues at the apply site if multiple changes occur on the same LOB. Additionally, LOBs can not be updated at the target sites.⁶ In contrast, Oracle Streams allows updates to LOBs at all sites. If an update to a portion or piece of a LOB column is captured, only those modified chunks of the LOB will be propagated and applied to the target sites.

⁴ DB2 Replication Guide and Reference, Version 8, page 185.

⁵ DB2 Replication Guide and Reference, Version 8, page 88.

⁶ DB2 Replication Guide and Reference, Version 8, page 99.

Staging and Propagation

Oracle	IBM
Oracle can route changes through multiple intermediate sites without applying the change at the intermediate site.	IBM applies changes directly to the target table.
Replication events and user messages can share staging and propagation infrastructure	Replication events and user messages must use different incompatible staging and propagation infrastructures

Table 3: Staging and Propagation Summary

Oracle Streams supports rule-based propagation of events (for example, Changes for all European sites are routed to a central European site and then distributed from there) and networked routing, for greater flexibility. Streams is designed for near real-time asynchronous replication although different scheduling can be applied for different target locations. Because events can be routed throughout a networked environment, virtually any configuration is supported by streams, including n-way connectivity in an update anywhere environment.

Oracle Streams captured data is initially staged (enqueued) at the source database. From here it can be applied locally or propagated to staging areas at one or more remote locations. Data in a staging area can either be applied or further propagated to another staging area. This allows for networked routing of events.

User messages can share Oracle Streams' staging and propagation infrastructure. Routing and staging rules need only be defined once, and are available to all applications.

Captured data in the IBM scenario can be staged at the source or in a middle tier. However, there is no capability to perform queue-to-queue propagation as in Oracle Streams. IBM's replication solution requires that both the source and target table be defined as part of the subscription set, there is no concept of routing changes destined for a remote site through multiple intermediate sites without applying the changes at the intermediate sites. In addition, replication and message queuing propagation and staging are incompatible and must be independently defined.

Rules and Subsetting

Oracle	IBM
Oracle Streams replication granularity extends beyond Subsets of tables and Tables, to include entire schemas or even the entire database.	IBM's replication is limited to Tables and Subsets of tables. ⁷
Tables, including row and column subsets, can be updated at the target site.	Table subsets cannot be updated at the target site. ⁸

Table 4: Rules and Subsetting Summary

With Oracle Streams, rules govern what events are applied at a particular location. Rules can be applied at three different levels: globally (for example, capture all changes in the database), by schema, and by table. A rule is specified as a condition that is similar to the condition in the WHERE clause of a SQL query, and an individual rule must evaluate to TRUE for the object to be selected. Users can group related rules together into rule sets. Thus, a rule can be used to perform column-level or row-level subsetting of data during capture, propagation, and/or apply. Each process uses the ruleset specified for itself to dequeue relevant change records from the queue. For maximum flexibility, Oracle Streams allows users to create rules, which are used to determine what data is captured, propagated and applied. Rules can be added to a running capture, propagation, or apply process without stopping the processes. The new objects will be added into the replicated environment automatically.

Oracle Streams supports both column-subsetting and row-subsetting during apply. The columns and column names do not have to match at the replicated site. In the case of row-subsetting, Streams automatically maintains the sites' data integrity of the subset with row migration. As an example, consider the following 3 sites. Site A maintains a complete set of employee records. Site B contains only Louisiana employee records, and Site C contains only California records. If an employee moves from the Louisiana office to the California office, then the record must be removed from the Louisiana site and inserted into the California site. This can be configured automatically with Streams so that the update from Site A is converted into a Delete statement at Site B, and an Insert statement at Site C. Column-level and row-level subsetting are fully supported by Oracle Streams for updateable replicas.

IBM does not support any subsetting for updateable replicas. IBM replication only supports column and row subsetting of read-only copies of a table. Column

⁷ DB2 Replication Guide and Reference, Version 8, page 37.

⁸ DB2 Replication Guide and Reference, Version 8, page 107.

subsetting must be specified when a table is defined as a replication source, or when a subscription set is defined.⁹ Row subsetting of the read-only target tables can only be defined during the definition of the subscription. Oracle Streams subsetting, in comparison, can be defined or modified at any time and does not limit the target site's ability to do real work.

Data Transformation and Apply

Oracle	IBM
Customizable apply capability for DML and DDL.	Apply DML changes only
Flexible transformation capability at each phase of replication: capture, propagation, and apply.	Limited transformation capability, only at subscription time, for read-only tables. ¹⁰

Table 5: Data Transformation and Apply Summary

With Oracle Streams, the apply process dequeues events from the local staging area and applies them to the local tables. Users have the option of using the default apply mechanism or, for greater flexibility, creating a custom apply mechanism. This custom apply capability gives the customer full control over the resulting replicated object, even allowing alternative SQL to be performed in place of, or in addition to, the original transaction's changes. These customized apply handlers can be specified and modified at the individual table level without stopping the apply process(es). DDL Apply handlers are specified at the apply process level, enabling centralized DDL administration for the specified tables. Additionally, users can elect to have their application explicitly dequeue events from the staging area.

Changes made during the Oracle Streams apply process by default are marked with an apply tag so that the capture process will ignore those changes. This prevents undesirable outcomes, such as replication cycles. Customers can modify this behavior, if desired, or set their own apply tags and rules to enable different outcomes.

Oracle Streams supports a variety of transformation during capture, propagation, and apply for updateable replicas. Transformations are frequently used to rename a column, schema, or table or change the datatype for a particular column for subsequent sites. Transformations are specified on rules for a particular object within a ruleset. If a transformation is specified for a rule, each logical change record matching the criteria of the rule is transformed as it is dequeued from the staging area, and that transformed logical change record is

⁹ DB2 Replication Guide and Reference, Version 8, page 42.

¹⁰ DB2 Replication Guide and Reference, Version 8, pages 88, 111.

passed to the appropriate process (capture, propagate, apply) for further processing. Transformations on capture result in the transformed change record being enqueued into the staging area and later propagated to all downstream sites. Transformations during propagation will impact any downstream site from the targetted staging area. Transformations during apply will only impact the individual target site. The customer has full authority and control over the data.

Additionally, non-Oracle databases can also be maintained using the Oracle Streams, apply process with a database link via Oracle Transparent Gateways. Transformations and Oracle custom apply procedures can also be used to manipulate the non-Oracle databases.

IBM supports limited transformations for read-only replicas only by using either a trigger, an expression through the subscription, or a source view .

The IBM model of apply is primarily pull-based, similar to Oracle’s materialized view technology. Oracle Streams uses a push-based model, which provides lower latency change propagation. IBM does not provide support for customized apply procedures for individual tables or objects.

Conflict Detection and Resolution

Oracle	IBM
Oracle provides multiple pre-built conflict resolution handlers for update conflicts as well as the ability to specify user-defined conflict resolution handlers.	The parent source table data always wins in an update conflict ¹¹

Table 6: Conflict Detection and Resolution Summary

Oracle Streams supports a much wider variety of configurations than IBM and also supports a much wider variety of conflict detection and resolution strategies, including overwrite, discard, maximum, minimum, and user-supplied.

With Oracle Streams, the apply process detects update, delete and uniqueness conflicts. Streams, automatically sends the before and after image of the requisite columns and changes in the logical change record. Oracle supplies a variety of pre-built conflict resolution handlers for update conflicts, including: overwrite, discard, maximum and minimum. Users can implement timestamp conflict resolution as a specialized case of the maximum pre-built conflict handler. Users can also write their own conflict resolution handlers for each of these three types of conflicts. In the event that a conflict cannot be resolved for a

¹¹ DB2 Replication Guide and Reference, Version 8, page 57.

particular transaction, the transaction is logged to the error queue and can be reapplied after the situation causing the error is resolved.

IBM provides three levels of conflict resolution: none, standard and enhanced. If “none” is selected, conflicts are not detected or resolved. Standard conflict resolution checks primary key values to determine if a change has been made at both the source and target tables. If so, the incoming change, from the replicated source, is discarded. With their enhanced conflict option, the Apply program locks all replicas in the subscription set against further transactions, and begins detection after all changes prior to the locking are captured¹². Enhanced resolution can only be used in a continuously connected environment and has a significant impact on user availability. Rejected changes are handled at the conclusion of the subscription set for apply, unlike Oracle Streams which handles conflicts on a transaction by transaction basis.

Configurations

Oracle Streams supports virtually any replication configuration; whereas IBM supports point-to-point replication. Because Oracle Streams events can be routed throughout a networked environment, replication models, including hub-and-spoke and n-way connectivity, are available in an update anywhere environment.

IBM replication is primarily read-only. Updateable replicas cannot have any subsetting, transformations or LOBs. IBM replication most closely resembles Oracle’s materialized views. IBM supports the concept of a single updateable source table (master) with multiple read-only or updateable target tables (materialized views).

Manageability

Oracle	IBM
Oracle Streams gracefully handles table structure and DDL changes while actively replicating changes.	Replication must be stopped when any DDL change is performed. ¹³
Oracle Streams is an integrated feature that functions identically on all hardware platforms.	Replication functionality and behavior is dependent on hardware platform, and may require installation of additional software packages ¹⁴ .

Table 7: Manageability Summary

¹² DB2 Replication Guide and Reference, Version 8, page 57.

¹³ DB2 Replication Guide and Reference, Version 8, page 185.

¹⁴ DB2 Replication Guide and Reference, Version 8, Chapters 17-21.

Oracle Streams, replication supports replication of both DML, as well as DDL changes without requiring any downtime. This allows users to add or drop columns from a replicated table, or to add or drop tables from a replicated environment while still applying and propagating DML changes. IBM does not support automatic replication of DDL changes.

Oracle replication is part of a unified Streams environment, which is fully integrated with the Oracle Server. There are no additional components to install or configure. Certain IBM platforms require the use of add on technology, most typically DB2 DataPropagator, to support replication. There are different versions of DataPropagator for different platforms (for example, DataPropagator for Z/OS, DataPropagator for OS/390, and so on). DataPropagator does not function identically across these platforms.

DETAILED COMPARISONS FOR MESSAGE QUEUING

Enterprises have developed autonomous and distributed applications to perform individual business tasks. These applications need to coordinate the business tasks in a consistent, reliable, and secure manner while maintaining their flexibility. Certain information such as customer and business partner communications need to be reliably audited for future reference or legal reasons. E-business evolutions require this level of coordination even among inter-business applications.

Applications coordinate with each other and at the same time store vital information about business tasks in a database. Oracle offers database integrated message queuing functionality. Using this functionality, message queuing operations such as enqueues and dequeues can be performed along with SQL operations in the same database transaction. This is a fundamental functionality essential for today's world.

IBM offers WebSphere MQ family of products (formerly called MQSeries family of products) for application coordination. These products essentially use WebSphere MQ (formerly called MQSeries) for communicating between applications. IBM has realized the importance of database integration for applications coordination. It has added table functions to Websphere MQ in DB2. It has also added a set of functions in WebSphere MQ Integrator (formerly called MQSeries Integrator). However, their integration is superficial and suffers from a variety of problems including performance and recovery problems. Table functions don't do any transaction coordination between Websphere MQ and DB2.

The following sections compare the functionality of Oracle Streams-based message queuing versus IBM Websphere MQ in the following categories:

- Installation and configuration
- Key Features

- Manageability
- Performance

Installation and Configuration

Message queuing is a functionality of the Oracle database. When you install the Oracle database, its message queuing functionality is already there and ready to use.

With IBM, you need to install DB2 and WebSphere MQ family of products – WebSphere MQ, WebSphere Workflow, and WebSphere MQ Integrator. Then you need to integrate WebSphere family of products and DB2. Integration of all these products is in itself a big problem.

Key Features

Integration Model

Oracle	IBM
Integrated Model	Inconsistent Model
<ul style="list-style-type: none"> • Hub-and-spoke model for integration 	<ul style="list-style-type: none"> • Point-to-point model for integration
<ul style="list-style-type: none"> • Integrated security for database and message queuing operations 	<ul style="list-style-type: none"> • Different security for database and message queuing operations
<ul style="list-style-type: none"> • Database and message queuing operations can be performed in the same transaction 	<ul style="list-style-type: none"> • Complex transaction coordination is required for message queuing and database operations
<ul style="list-style-type: none"> • Message queuing use the same data structures as database operations 	<ul style="list-style-type: none"> • Limited type system for message queuing operations.
<ul style="list-style-type: none"> • In case of failures, recovering database recovers both SQL and message queuing operations 	<ul style="list-style-type: none"> • Recovering WebSphere MQ is different from recovering DB2.

Table 8: Integration Model Summary

Oracle offers the ability to integrate messages in the Oracle database. Multiple applications can communicate using a single queue in an Oracle database. Message queuing operations can be performed in the same transaction as SQL operations without requiring distributed two-phase commit. Messages can be of

any arbitrarily complex user-defined data type. Oracle database security can be utilized for the message queuing operations.

IBM has fundamentally point-to-point transport-based integration model. Each application has its own local MQ queue. These queues are the communication channels between applications. If a third application would be added, a third queue would need to be created and connected to the other two queues. This creates a big management headache and is the source of all the issues with WebSphere MQ.

In IBM's solution, message queuing and database operations are two totally different products. Merely providing table function access to WebSphere MQ – only on AIX 4.3.3 – does not solve the problem. There remains the additional complexity of dealing with other things such as security and distributed two-phase commit. The data model for DB2 is different from WebSphere MQ. The security model of DB2 is different from WebSphere MQ. In case of failures, transactions have to be recovered from both DB2 and WebSphere MQ. Known problem of indeterminate state in case of failures remains with their use of distributed 2-phase commit.

Message Payload type

Oracle	IBM
Extensible type system for message queuing operations.	Limited built-in type support via Websphere MQ
Complex types can be designed using Oracle objects and XMLType functionality	Need to write your own parser for complex types

Table 9: Message Payload Summary

In the message queuing world, message structures can be very complex. Customers using Oracle's message queuing technology can take full advantage of the extensive type support of the Oracle database. Message queuing users and the database users create the user-defined type once and use it for both purposes. WebSphere MQ has limited built-in type support¹⁵. It only supports unstructured RAW and text messages. WebSphere MQ Integrator has added support for using any parser. Writing your own parser can be very complex. In addition, type support of DB2 is different from that of WebSphere MQ. As a result, IBM users have to manage an added complexity of managing different data structures at

¹⁵ MQ Series Application Programming Reference, Chapter 1: Data type Descriptions

two places. In addition, they have to manage the complexity of message transformation between the two structures.

Publish/Subscribe and Point-to-multipoint Communications

Oracle	IBM
Integrated publish/subscribe and point-to-multipoint communication	Publish/subscribe and point-to-multipoint built on top of point-to-point model
Same queue can be used for both publish/subscribe and point-to-multipoint	Multiple copies of messages have to be enqueued for point-to-multipoint Publish/subscribe requires a complex web of MQ channels and MQ streams

Table 10: Communication Model Summary

Oracle offers a unified publish/subscribe and point-to-multipoint communication models. A single queue can be used for both publish/subscribe and point-to-multipoint communication in an easy-to-use way. If a message is enqueued with a recipient list then the messages are delivered to the specified list of recipients. Otherwise, the message is delivered to the registered subscribers based on their rules. In Oracle’s model, both point-to-multipoint and rule-based publish/subscribe have same performance characteristics. Complex rules are evaluated using highly optimized SQL expression evaluation engine.

IBM offerings are based on inherently point-to-point communication technology. Their publish/subscribe is implemented on top of point-to-point model. It involves a complex web of MQ channels and MQ streams. As a result, it is not only very difficult to use but can’t scale with number of messages or number of queues in the system.

From IBM’s report¹⁶, we get the following numbers:

- 1K non-persistent messages with a single subscriber: 438 msgs/second
- 1K persistent messages with a single subscriber: 48 mgs/second

A study done with Oracle8.1.5 on similar hardware showed that the Oracle database, could process around 1500 messages (message size - 2K) per second per queue with both publish/subscribe and point-to-multipoint communications.

¹⁶ MQSeries Integrator for Sun Solaris V2, Performance report, <http://www-3.ibm.com/software/ts/mqseries/txppacs/supportpacs/ip64.pdf>

Message Gateway

Oracle	IBM
Interoperability with third-party message queuing systems such as Websphere MQ	No such functionality

Table 11: Message Gateway Summary

Oracle understands that there are legacy applications, which are only accessible via WebSphere MQ (formerly called MQ Series). Oracle supports message gateway to Websphere MQ. The message gateway is a feature of the Oracle database as of Oracle9i. It allows automatic propagation of messages from Oracle queues to a WebSphere MQ queue and from WebSphere MQ queue to an Oracle queue. The message gateway allows users to define message transformations that are automatically invoked during propagation.

Message Queuing Functionality

WebSphere MQ has limited message queuing functionality as compared to Oracle message gateway.

Oracle	IBM
Message can be enqueued with a specified delay	WebSphere MQ has no such functionality
Oracle supports both FIFO and priority-based ordering. Priority can be any Oracle number	Limited priority support ¹⁷ – priority has to be between 0 and 9. FIFO is not guaranteed for enqueues into remote queues ¹⁸
Oracle offers extensive notification functionality – PL/SQL, Java, OCI, Email, and HTTP Post notifications	Limited signaling support on limited platforms ¹⁹
Oracle offers integrated point-to-multipoint and publish/subscribe model	IBM's offerings are primarily based on point-to-point model. Publish/subscribe offering can't scale with number of messages or number of queues in the system
User-defined transformations can be used at various points in message queuing	SQL-based transformations offered in costly WebSphere MQIntegrator
Oracle offers SOAP-based access to its queues	There is no such functionality in IBM's offerings.
Streams supports automatic enqueue of messages via the capture process	All messages must be explicitly enqueued to WebSphere MQ
Streams supports automatic consumption of messages via the default or a custom apply process	All messages must be explicitly dequeued by WebSphere MQ

Table 12: Message Queuing Functionality Summary

Manageability

Manageability has two aspects:

- System management
- Message Management

¹⁷ MQ Series Application Programming Guide, page 32

¹⁸ MQ Series Application Programming Guide, page 126

¹⁹ MQ Series Application Programming Guide. Page 16

System Management

Oracle	IBM
Single tool – Oracle Enterprise Manager - to manage SQL tables and message queues	SQL tables and message queue need different tools for management
Database recovery recovers both SQL and message queuing operations	Message queuing operations and SQL operations need to be recovered separately

Table 13: System Management Summary

Oracle offers a single tool for managing message queues – Enterprise Manager. Enterprise manager provides the management capabilities for managing both tables and message queues. You can create, drop, and manage queues just as you can manage tables. In case of failures, you recover messages along with the table rows when you recover the Oracle database. You have all the Oracle system management functionality – recovery, high availability, RAC – for message queuing users. Diagnostics and tuning pack of Enterprise manager allows tuning and monitoring of queues.

With IBM, message queuing is separate from database. Message queuing has to be managed separately from the database. Recovery of WebSphere MQ is different from recovery of DB2. In this case, you need to deal with system management of four different products – DB2, WebSphere MQ, WebSphere Workflow, and WebSphere MQ Integrator.

Message Management

Oracle	IBM
Since, all the messages in the queues are accessible via SQL, single SQL query can be written to find out exact message state	Messages are scattered in different queues at different sites. Each site has to be queried to find the exact message state ²⁰
Message queuing operation is automatically audited and history is retain for specified duration	Auditing logic of message queuing operation has to be written in application

Table 14: Message Management Summary

Applications can communicate using queues in an Oracle database. Applications producing the message put messages in a queue. The applications consuming the

²⁰ MQ Series Application Programming Guide. Page 61

messages consume the messages from the same queue. Applications on different Oracle databases can use the queue-to-queue propagation feature to propagate the messages between queues in different databases. All the messages in a queue are also accessible via a SQL view. Messages are retained after consumption, thus available for auditing. The Oracle database is a single repository for all the messages. These features allow for easy tracking and management of messages.

With IBM, each application has its own local queue and queue manager. Applications communicate with each other using local and remote queues. Queue managers exchange messages using distributed the two-phase commit protocol. There is no single repository for the messages. Every queue has to be searched to find a message. IBM customers always run into the problem of “where is my message?” IBM does not store messages after consumption. In order to keep track of messages for future reference a database has to be used. IBM does not offer any automatic auditing capabilities like Oracle. Database operations have to be coordinated using distributed two-phase commit, adding an extra layer of complexity.

Performance

Oracle	IBM
Message queuing operations benefit from the performance optimizations of Oracle database and hence has proven to be much faster than Websphere MQ	Websphere MQ can't utilize performance optimizations of DB2
Scalability features of Oracle database can be used for scaling message queuing operations	Scalability of DB2 has no impact on the scalability of Websphere MQ

Table 15: Performance Summary

Message queuing operations in Oracle make use of performance techniques in the Oracle database such as buffer management and indexing techniques. Buffer management allows optimized access to the messages when both consuming and producing applications are on the same database. Consuming applications need not fetch the message from disk if the message is still available in the buffer cache. Indexing techniques allow for faster access of the consuming application's messages. Message queuing applications also benefit from optimized log writing. In addition, there is no distributed two-phase commit involved while performing SQL operations along with message queuing operations.

With IBM, WebSphere MQ and DB2 are two separate products. As a result, WebSphere MQ can't utilize the performance techniques built in DB2. WebSphere MQ maintains its own logs while DB2 its own. Transaction

coordination between DB2 and WebSphere MQ requires distributed two-phase commit protocol. As a result, the performance of WebSphere MQ family of products is lagging way behind Oracle's message queuing functionality.

From IBM's report²¹ we get the following numbers:

- 1K Non-persistent messages with single MQInput and MOutput node: 1318
- 1K persistent messages with single MQInput and MQOutput node: 56
- 1K non-persistent messages with database node: 46
- 1K persistent messages with database node: 23

This performance benchmark was conducted on Sun E450 with 4*400MHz processors as server. These numbers highlight the following facts:

- Drastic degradation in the performance of WebSphere MQ by simply adding recovery logs to non-persistent messages
- Degradation in the performance due to distributed two-phase commit in the database node.

A study conducted with Oracle 8.1.5 on Sun Solaris E450 with 4*400MHz processors showed that the Oracle database can process around 1500 messages per second per queue. The study was conducted with 2K persistent messages. Note that distributed two-phase commit is not required for SQL operations with Oracle. Performance has been further enhanced in later releases of the Oracle database.

DETAILED COMPARISONS FOR SYNCHRONOUS DATA ACCESS (FEDERATED DATABASE)

Oracle9i is uniquely scalable with technologies like Real Application Clusters. This enables it to support customers' largest data stores in a single database, when other vendor's require multiple databases. This has often led to a perception that Oracle does not have strong distributed synchronous data access features. This is hardly the case.

Oracle has provided all the key features important for distributed synchronous data access for many years. Oracle introduced support for distributed queries and location transparency in Version 5.1 of the Oracle database. Distributed transactions and support for heterogeneous sources were introduced in Version 6 of the Oracle Database. Support for distributed SQL and transactions are inherent to the Oracle database, no other product is required.

²¹ , "MQSeries Integrator for Sun Solaris V2, Performance report", <http://www-3.ibm.com/software/ts/mqseries/txppacs/supportpacs/ip64.pdf>

Oracle's support for the heterogeneous distributed environment enables, Oracle clients to use Oracle SQL to access and manipulate data, that is physically located in a non-Oracle data store using a single SQL statement. Oracle has two solutions for transparently accessing non-Oracle data sources. They are Generic Connectivity and Oracle Transparent Gateways. These two solutions make it possible to access any number of non-Oracle systems from an Oracle environment in a heterogeneously distributed environment. Both of these are based on the Heterogeneous Services module of the Oracle database.

The Oracle database is the engine and front end, for providing heterogeneous connectivity. It coordinates all the transactions and maintains data integrity. The capabilities of the non-Oracle data source, is stored locally in the Oracle database. This eliminates the need for the Oracle server to fetch this information each time a connection is established with the non-Oracle system, resulting in a reduction of round-trips and data transferred necessary, to establish the connection with the non-Oracle system, thus enhancing performance.

Generic Connectivity is Oracle's low-end solution for accessing non-Oracle data sources using industry standards such as ODBC and OLE DB. Oracle Transparent Gateways are Oracle's tailored solution. They are built specifically for a particular data source and communicate with the data source using its native interface.

Both of these solutions fully exploit all the features of the Oracle database since it serves as the engine. The highly efficient and powerful SQL parser and optimizer of the Oracle database are two such examples.

Oracle refers to the ability of combining information from multiple data sources as distributed computing. IBM on the other hand uses the term 'Federated Database' to refer to this same concept. IBM has been heavily marketing its new feature in DB2, support for heterogeneous distributed computing. Distributed computing is critical to IBM, due to their lower scalability limits in a single database. According to IBM, *"In the 1990s, distributed computing began to offer the potential to radically transform our businesses. This transformation has led to the expectation that distributed computing will support the interconnection of a rich mix of heterogeneous components into a single, integrated information system solution."*²² Although IBM is clearly stating that distributed computing is an important feature of the database, it is now that they are adding support for it. Up until DB2 v7, IBM did not have any support for heterogeneous distributed queries and heterogeneous distributed transactions are still not supported. Support for heterogeneous distributed transactions is critical for customers seeking to support a single information system solution.

²² Database Integration With DB2 Relational Connect Building Federated Systems with Relational Connect and Database Views , Stephen Rutledge, John Medicke, 2001

IBM's model for integrating information in a distributed environment, so far, has been with their middleware product called DataJoiner. Even IBM agrees that this is not the best way to integrate data from many different sources. As a result, one of their major focus areas in Version 7 of the DB2 UDB product line was tighter technology integration. *A subset of functionality available in DataJoiner is incorporated into DB2 UDB v7.*²³

*"May 8, 2001 -- IBM announced new DB2 database software, which enables customers to integrate data from a wide variety of sources in the industry, including Web application integration, access to critical data sources and legacy applications. DB2 Version 7.2 will be generally available worldwide on June 8, 2001."*²⁴

DB2 Relational Connect is a product that is still in an infancy stage. While Oracle customers have been able to perform heterogeneous queries since 1986, DB2 UDB did not support heterogeneous distributed SQL without purchasing, an extra product, DataJoiner until 2001. Heterogeneous distributed transaction support is still non-existent in DB2 UDB.

The table below provides a comparison summary of the key features offered by Oracle and IBM DB2 with Relational Connect in the synchronous data access area. It shows that Oracle provides all the key features required to support distributed heterogeneous databases, including some that are not supported by IBM.

²³ C12 – DB2 UDB Technical Directions for this New Millennium, Goerge Baklarz, October 2000

²⁴ IBM: Attempting to Make DB2's Presence More Universal -- M. Schiff looks at DB2 UDB V7.2.

Feature	Oracle	IBM
Integrated Solution	Yes	No
Distributed SQL	Yes	Yes
Optimization of Distributed Queries	Yes	Yes
Heterogeneous Distributed SQL	Yes	Yes
Location Transparency	Yes	Yes
Heterogeneous Distributed Insert, Update and Delete Support	Yes	Yes ²⁵
Heterogeneous Distributed Transaction Support	Yes	No

Table 16: Key Synchronous Data Access Features

The following sections compare Oracle's offerings for the distributed environment, with IBM's DB2 Relational Connect in the following categories:

- Installation and configuration
- Key Features
- Performance

Installation and Configuration

Oracle's solutions for heterogeneous synchronous data access is integrated with the Oracle database, thus making installation very easy. Generic Connectivity is a component of the database. It is installed by default with the database, requiring no separate install. This removes all compatibility issues that customers face when trying to integrate different products.

Like Generic Connectivity Transparent Gateways, are also tightly integrated with the database. They are however not installed by default with the database. They can be installed as a stand-alone component, thus providing the flexibility to be installed on any machine in the network. Customers can choose to install it on the same machine where the Oracle database is installed, or on the machine where the non-Oracle data source is installed, or on any other machine. Such flexibility provides customers the choice to install the product where it is most appropriate for their environment. As is the case with Generic Connectivity there are no compatibility issues.

Contrast this single step installation with IBM's offering. DB2 Relational Connect is a separate product from DB2 UDB. It is installed separately from

²⁵ Only supported with DB2 family of products and Informix data sources. P. 75 IBM DB2 Universal Database: What's New

DB2 UDB. It only works with DB2 Universal Database EE or EEE²⁶ and requires the DB2 Universal Database to be configured as a Federated system to enable distributed access.

In contrast to Oracle Transparent Gateways, DB2 Relational Connect does not offer the flexibility to be installed on any machine on the network. It has to be installed on the same machine as the database.

Lastly, IBM requires customers to determine at install time whether or not their database will be used in a federated manner. Should they not install the federated libraries, the database cannot later access other databases.²⁷

Data Sources Supported

Oracle Transparent Gateways provide access to Sybase, Microsoft SQL Server, Informix, DB2, Teradata, Rdb, Ingres and RMS. In addition, Generic Connectivity enables Oracle to access any data source that supports an ODBC or OLE DB driver. Oracle even supports access to file systems, such as Excel, through OLE FS.

In contrast, DB2 Relational Connect supports access to only a few major database vendors. *“In version 7, DB2 Relational Connect configures access to Oracle, Sybase, Microsoft SQL Server, and DB2 data sources.”*²⁸

DB2 v8, only supports access to DB2 family of products and Informix. *“If you are using DB2 Universal Database Version 7.x to access any other data sources, it is recommended that you upgrade when the new information integration functionality is available. If you decide to upgrade to DB2 Universal Database Version 8.1, you will be able to access only the DB2 family of products and Informix data sources. The wrappers to any other data sources will not work.”*²⁹

Platform Availability

Oracle supports far more platforms than IBM. Generic Connectivity is available on Solaris, NT, HP and AIX. Transparent Gateways are available on numerous platforms:

- Transparent Gateway for MS SQL Server is available on NT
- Transparent Gateway for Informix is available on Solaris, HP/UX
- Transparent Gateway for Ingres is available on Solaris, HP/UX

²⁶ DB2 Relational Connect, <http://www-3.ibm.com/software/data/db2/relconnect/index.html>

²⁷ <http://www-3.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/document.d2w/report?fn=db2v7iydb2iy143.htm>

²⁸ DB2 Relational Connect, <http://www-3.ibm.com/software/data/db2/relconnect/about.htm>

²⁹ <http://www-3.ibm.com/software/data/db2/relconnect/migrate.html>

- Transparent Gateway for Teradata is available on Solaris, NT, HP/UX
- Transparent Gateway for RDB is available on Alpha OpenVMS
- Transparent Gateway for RMS is available on Alpha OpenVMS
- Transparent Gateway for Sybase is available on Solaris, HP/UX, NT, AIX, Tru64
- Transparent Gateway for DB2 on OS390, AS400
- Transparent Gateway for DRDA on Solaris, NT, AIX, HP

Relational Connect is much less platform friendly. “*DB2 Relational Connect runs on AIX, Linux, Solaris and Windows NT/2000.*”³⁰

Key Features

Distributed SQL

Oracle has supported Distributed SQL since 1986. Users and applications have been able to access and manipulate data that resides in a distributed data source using a single SQL statement, since Oracle database Version 5.1.

An Oracle database can be configured to access numerous other databases. Such a database will contain all the necessary data dictionary information about the other remote databases so it can access and understand their characteristics.

IBM has attempted to differentiate itself as a leader, by creating a new term for Distributed SQL. DB2 v7 introduced a major new feature, ‘Federated Databases’. What exactly is a ‘Federated Database’? According to the IBM documentation a federated database system is “a database management system that supports two or more databases in a single statement.”³¹ In other words, a Federated Databases is nothing more than Distributed SQL, a feature Oracle has provided for 15 years.

Heterogeneous Distributed SQL

As an extension of the distributed SQL feature, Oracle has support for heterogeneous sources. With this feature users and applications can access and manipulate data that resides in an Oracle as well as a non-Oracle data source using a single SQL query, in Oracle syntax.

Oracle has Transparent Gateways for all the major relational databases in the industry. In addition to that, Oracle provides Generic Connectivity. This is a database feature that enables users and applications to access data residing

³⁰ <http://www-3.ibm.com/software/data/db2/relconnect/requirements.html>

³¹ P.53 IBM DB2 Universal Database Administration Guide: Planning

anywhere using industry standards such as ODBC and OLE DB. This feature even supports access to data residing in file systems using OLE FS.

DB2 v7.x only supports a few selected databases, Oracle, Sybase, Informix and SQL Server. IBM is positioning federated databases as a way to “*build a highly integrated, consistent system that exploits the full power of business intelligence through the DB2 SQL interface using DB2 Relational Connect.*”³² So what if your organization has data that resides in other data sources such as RMS? Since probably about 40% of the data in the world is not stored in Oracle, Sybase, SQL Server, Informix or DB2, there is a high probability that will be the case. What then?

DB2 v8, only “*has the ability to federate relational data across IBM’s family of databases, including DB2 and Informix™ IDS.*”³³

Location Transparency

Location transparency enables Oracle users and applications to reference objects that reside in any remote data sources, but the location of the objects is transparent to the user. The user refers to the objects using synonyms or nicknames. A synonym is a user-defined, alias for a table, view, sequence, or program unit. A synonym is not actually a schema object itself, but instead is a direct reference to a schema object. Synonyms can be defined for Oracle or non-Oracle data sources.

Since location transparency separates the location of a data source from its physical location, in the event that any object being referenced, is moved, the calling application does not have to change. Only the synonym definition will have to be changed.

Oracle databases have supported location transparency since Version 5.1. IBM introduced location transparency in their database only about 3 years ago. Prior to that they required DataJoiner. Location transparency is an extension of distributed SQL, since DB2 did not support distributed SQL till recently they could not support location transparency.

Distributed Update Support

Oracle provides complete heterogeneous data management with SELECT, INSERT, UPDATE, DELETE. Both Generic Connectivity and Transparent Gateway have the ability to transparently read and write to a non-Oracle source. Data that resides in the non-Oracle source can be transparently queried and updated using Oracle SQL syntax. Existing data can be changed and new data can be inserted or deleted from the non-Oracle source.

³² DB2 Relational Connect <http://www-3.ibm.com/software/data/db2/relconnect/index.html>

³³ P.75 IBM DB2 Universal Database: What’s New

“DB2 Relational Connect provides native read access to Oracle, Sybase, and Microsoft SQL Server databases.”³⁴ This means that the data that resides in the non-IBM sources can only be queried. It cannot be changed.

Distributed Transaction Support

Oracle’s database has had support for distributed transactions for many years. The ability to update data in two or more database(s) using one or more statements in a transaction has been a feature of the Oracle database since Version 6. Oracle guarantees data integrity in a distributed transaction by guaranteeing that all database servers participating in a distributed transaction either all commit or all roll back the statements in the transaction. Oracle's two-phase commit mechanism provides this guarantee.

Oracle extends its support for distributed transactions to non-Oracle data sources as well. Oracle Transparent Gateways have full support for the two-phase commit mechanism thus ensuring data integrity and consistency with non-Oracle data sources.

Oracle supports distributed transactions with non-Oracle data sources even if the non-Oracle data source does not support it. When a distributed transaction involves an Oracle database and a single non-Oracle data source, Oracle can compensate for the shortcomings of the non-Oracle data source and still ensure data integrity using the two-phase commit mechanism.

DB2 on the other hand, does not support distributed transactions in a heterogeneous environment. Therefore for a DB2 customer to be able to update more than one database in a single transaction, they would have to use DataJoiner, IBM’s middleware product.

Optimization of Distributed Queries

The Oracle database stores information about all the remote databases that it can access. For queries referencing objects in the remote database, the Oracle optimizer will use this information to determine the most efficient way to execute a SQL statement, since there are often many different ways to execute a SQL statement; for example, by varying the order in which tables or indexes are accessed.

The procedure Oracle uses to execute a statement can greatly affect how quickly the statement executes. Oracle's cost-based optimizer can transparently rewrite, distributed queries to take advantage of the performance gains for example gains, offered by collocated inline views.

For example, if there is a query that joins three tables, one in Oracle and two tables in a remote database, the Oracle optimizer will perform the join of the two

³⁴ DB2 Relational Connect, <http://www-3.ibm.com/software/data/db2/relconnect/index.html>

remote tables on the remote table and then return the results to Oracle where the join with the Oracle table will be performed. This reduces the number of rows that is moved across the network thus enhancing performance of the query. If the tables on the remote system were huge, for example, 1 million rows each, then moving all the rows across the network would be very costly. In such a case the optimizer determines that it is less costly to perform the join on the remote system. Performing the join on the remote system greatly reduces the number of rows moved across the network, thus improving the performance.

Materialized Views

Oracle provides full support for summarized data of remote sources, referred to as Materialized views. The materialized views can be read only or updateable.

The Oracle optimizer can use these materialized views to improve query performance by automatically recognizing when a materialized view can and should be used to satisfy a request. The optimizer transparently rewrites the request, to use the materialized view. Queries are then directed to the materialized view and not to the underlying detail tables or views thus enhancing the performance of the query.

Performance

IBM has made many claims about its ability to access data in a distributed environment faster than Oracle. However they have yet to produce any proof to substantiate their claim. Since these are just IBM's words, until such time that IBM produces proof of such a performance superiority Oracle will file this claim as fiction and not fact.

CONCLUSION

Oracle offers all the features required for information integration in a single unified product. The inherent reliability and security features of the Oracle database are automatically inherited by information integration applications.

IBM on the other hand has a completely different model for information integration. They offer different products for different integration scenarios. This adds several layers of complexity, since each product has to be installed and configured separately.

Oracle is clearly the leader in the distributed space with its single, unified solution to satisfy a complete spectrum of information integration requirements.



White Paper Title

January 2003

Author:

Contributing Authors:

Oracle Corporation

World Headquarters

500 Oracle Parkway

Redwood Shores, CA 94065

U.S.A.

Worldwide Inquiries:

Phone: +1.650.506.7000

Fax: +1.650.506.7200

www.oracle.com

**Oracle Corporation provides the software
that powers the internet.**

**Oracle is a registered trademark of Oracle Corporation. Various
product and service names referenced herein may be trademarks
of Oracle Corporation. All other product and service names
mentioned may be trademarks of their respective owners.**

Copyright © [Year] Oracle Corporation

All rights reserved.