ORACLE STREAMS FEATURE OVERVIEW

ORACLE STREAMS USES

ORACLE STREAMS FLEXIBLE ARCHITECTURE SUPPORTS MANY IMPLEMENTATIONS

- Create a reporting site and offload processing from an OLTP site.
- Provide load balancing and improved scalability and availability for a call center or similar application.
- Provide site autonomy between locations to satisfy certain common business requirements.
- Integrate multiple, best-of-breed applications.
- Provide fast, local access to data at locations with indirect connectivity, such as remote sales locations.
- Transform and consolidate data from multiple locations, such as regional offices.
- Provide event notification and work flow.
- Upgrade an application or platform with virtually no down time.

Oracle Streams provides a flexible infrastructure that meets a wide variety of information sharing needs. Oracle Streams enables the propagation of data, transactions and events in a data stream either within a database, or from one database to another. The flexibility of Streams over traditional solutions for data replication and message queuing, allows customers to select a single information sharing solution, and to deploy information solutions in less time and for less cost. As users' needs change, they can simply implement a new capability of Oracle Streams, without sacrificing existing capabilities.

Overview

Oracle Streams provides a set of elements that allow users to control what information is put into a stream, how the stream flows or is routed from node to node, what happens to events in the stream as they flow into each node, and how the stream terminates. There are three distinct phases to an Oracle Streams environment: capture, stage, and consume (apply). Depending upon how these phases are implemented, Oracle Streams can be used to address a wide variety of requirements including n-way replication, hub-and-spoke replication and message queuing.

Load Balancing with N-Way Replication

An n-to-n or multimaster model is often used to share work between two or more locations. By load balancing between multiple sites, companies can provide improved scalability and availability of data to their end-users. This type of configuration is particularly useful in query intensive applications, such as an online order entry system. Customers generally view multiple products before placing their order. Because only the actual update would need to be propagated between multiple locations, the overhead of replicating transactions is minimal, compared to the benefit of distributing the query load. As the business grows, it is easy to add additional replicas to provide additional scalability. These replicas can be co-located, or may be geographically distributed to provide improved response time for each region. Sites can be added to a configuration without affecting the availability of the existing sites—there is no down time.

Each site has fast, local access to the data, and should a site become unavailable for any reason, transactions can be routed to one of the surviving locations. The replication is completely transparent to the end-user. As updates (both DML, as well as DDL) are made to a table, they are automatically captured by Oracle Streams,

propagated to one or more designated locations, and automatically applied to the replicated tables in a near realtime manner.

Because updates to the same data are allowed at more than one location conflicting updates can occur. Oracle Streams automatically detects any conflicts and can resolve them in a consistent manner.

In a call center application, it might make sense for all sites to contain a full copy of all the data. However, this is not a requirement of Streams. You can choose to replicate an entire database, a particular schema, one or more tables, or even a subset of a table depending upon your business needs. Each replica may look completely different from the other replicas, and you may choose to transform data as it is replicated from one location to another.

Information Dissemination with Hub-and-Spoke Replication

It may not be desirable, or even possible, for all sites in your replication environment to be connected with one another. Companies frequently require an easy method of disseminating information to multiple locations. In this sort of 1-N, or hub-and-spoke configuration, changes made at the primary or hub location can be propagated out to the remote or spoke locations in a near real time manner.

Although it is possible to configure a hub-and-spoke configuration for bi-directional replication, many customers prefer to restrict updates to a single location, the hub. In query intensive environments, the load can still be balanced between multiple locations, with fast local access, while updates are restricted to the hub. By offloading reporting to the spoke locations, performance is also improved at the hub, or primary OLTP location. This type of configuration is easier to implement than multi-master replication as it is not necessary to establish connectivity between all locations in the replication environment, and it is not necessary to implement a conflict resolution strategy.

Hub-and-spoke replication with read-only spokes is easily implemented using the wizards provided by Oracle Enterprise Manager, or by making a single call to the appropriate MAINTAIN_* procedure in the DBMS_STREAMS_ADM package. Once configured, no further action, beyond routine monitoring, is required. DML and DDL changes preformed at the hub are automatically captured and propagated to the designated locations, where they are automatically applied.

The same procedure can be used to implement a 1-N configuration with downstream capture, which is useful to offload overhead from the primary site, as well as to implement an N-1 configuration, which can be used to consolidate information from multiple sites to a single database. Additional information on using the MAINTAIN_* procedure to configure either uni-directional or bi-directional hub-and-spoke replication can be found in the *2day+ Data Replication and Integration Guide*.

Table Replication with Synchronous Capture

If you need to capture changes to only a small number of tables in your database, you may find it more efficient to use synchronous capture. This feature also allows users of Oracle Database Standard Edition to capture table changes. Instead of asynchronously mining the redo log, a synchronous capture uses a highly efficient internal mechanism to capture data manipulation language (DML) changes when they are made to tables. These changes can then be transformed as desired, propagated and applied at one or more locations.

Message Queuing with Oracle Streams Advanced Queuing

Using Oracle Streams Advanced Queuing (AQ), a rich message queuing feature integrated with the Oracle database, applications can securely and reliably communicate with one another in an asynchronous manner. Oracle Streams Advanced Queuing supports all the standard features of message queuing systems including multi-consumer queues, publish and subscribe, persistent and buffered queues, content-based routing, Internet propagation, and transformations.

One of the most common uses of information integration is for distributed applications to communicate with each other to coordinate business processes and tasks in a consistent, reliable, and autonomous manner. For example, an application developer can use Oracle Streams AQ to allow the shipping department to easily notify the billing department when a product has shipped, so that the customer can be billed accordingly. By combining Oracle Streams with the appropriate messaging gateway, applications can even interoperate with other message queuing systems, such as Tibco Rendezvous or IBM Websphere MQ. This ability can be especially useful when it is necessary to share information with business partners or customers with non-database oriented messaging systems.

Like all Oracle Streams implementations, information must be captured, staged and consumed. In the case of messages however, enqueue, stage and dequeue would be the more commonly recognized terminology. With replication, changes can be automatically captured and applied. With Oracle Streams AQ, application developers must provide the code to explicitly enqueue and dequeue the messages. Once enqueued, messages can be transformed and propagated as desired, before being dequeued.

Application developers can access Oracle Streams AQ through the following interfaces: PL/SQL, Visual Basic Java Message Service (JMS), and HTTP(S)

KEY FEATURES

SINGLE, UNIFIED SOLUTION

- Satisfy all information sharing requirements with a single solution
- Share data between both Oracle and non-Oracle datastores

INTEGRATED FEATURE OF ORACLE DATABASE

- No additional software to install
- Take advantage of reliability and security of Oracle Database

FLEXIBLE

- Capture DML and DDL
- Rule-based filtering
- Transformations
- Networked routing

MANAGEABLE

- Oracle Enterprise Manager wizards and monitoring
- Simplified configuration with single API call
- Performance monitoring tools

PERFORMANT

- Hot mining of online redo log minimizes latency
- Downstream capture reduces load on key sites
- Parallel capture and apply processes ensure maximum throughput for concurrent events.

Key Features of Oracle Streams

Flexible

- Message queuing
- Bi-directional and uni-directional replication
- N-way, hub-and-spoke, 1-N, N-1 and networked configurations
- Automatically capture DML as well as DDL changes
- Filter captured changes based on user-supplied rules
- Route data through staging areas before applying
- Declarative and user-defined data transformations

Manageable

- Single-step API's simplify configuration
- Oracle Enterprise Manager Wizards provide ease-of-use
- Oracle Streams Performance Advisor recommends performance optimizations

Performant

- Hot mining of online redo logs minimizes latency
- Downstream capture reduces load on critical locations
- Parallel capture and apply processes ensure maximum throughput for concurrent events

Single, Unified Solution

Oracle Streams is an integrated feature of the Oracle Database; there is no additional software to install. This allows the Streams environment to take advantage of all of the standard reliability and security features of the Oracle Database. Oracle Streams satisfies the most demanding information sharing requirements using a common infrastructure. Complex distributed environments benefit from a single solution that satisfies all their information sharing requirements. As an organization grows, developers and administrators can be confident that Oracle Streams has the flexibility to meet their changing requirements.

ORACLE