HVR

DATASHEET

Oracle Database Support

The Oracle Database is one of the most commonly used databases, running some of the world's most demanding and mission-critical workloads. The Oracle Database is a common data source for analytical environments and data lakes. This datas heet describes HVR's support for the Oracle Database as a source for change data capture, and as a target for data delivery.

Version Compatibility

HVR supports many Oracle Database versions and deployment options:

- 9.2 and above for log-based Change Data Capture, up to current
- 8.1.7 and above, up to current, as source for one time data load (refresh), and as a target for change delivery
- Single instance and clustered (RAC) deployments (including Oracle Exadata, using file systems to store transaction logs and Oracle's Automatic Storage Manager (ASM), or raw devices)
- Pluggable and non-pluggable databases
- Enterprise, Standard and Express Editions
- Transparent Data Encryption for all different ways the Oracle Wallet can be configured
- Oracle Data Vault
- Connectivity to the Oracle Database is established through Oracle's native drivers using TNS

Why HVR?

To support Change Data Capture from extremely high volume, mission-critical workloads running on the Oracle Database, you need an efficient, robust and flexible CDC solution. Beyond Oracle's solution, that primarily focuses on the Oracle ecosystem, HVR is that solution, whether your need is for heterogeneous replication scenarios, or as a more cost-effective alternative in the Oracle to Oracle world. HVR provides not only CDC capabilities but also includes:

- Target table creation automatically mapping data to compatible data types
- Refresh i.e. one time data load, integrated with CDC to simplify initial load scenarios
- Compare and repair, also in heterogeneous environments
- Rich statistics visualization
- Automatic monitoring
- Graphical User Interface

One-time Load – Refresh – and Compare/Repair

One-time load, in HVR's terms refresh, as well as compare/repair, starts with a select from the Oracle Database. Parallelism can be specified across tables, and data is pulled out through a native Oracle connection. For optimum performance and efficiency HVR recommends an architecture using an agent close to if not on the database server(s) to densely compress/decompress the data, and optionally encrypt/decrypt it. The native connection combined with optimized network utilization supports efficient and fast data retrieval.

For source Oracle HVR provides an option to run a flashback query as of a timestamp or SCN (System Commit Number) to enable an explicit point-in-time view of the data across multiple tables, to ensure the initial load starts with a consistent data set.

For target Oracle HVR uses efficient direct path data loads for optimum load performance. HVR will automatically disable referential integrity constraints until a transactionally consistent state is reached, or with deferrable constraints on the target tables HVR can combine a transactionally consistent select with a single consistent load on the target that only requires constraint validation to be deferred.

CDC from Oracle

The Oracle Database records changes in the transaction log in commit order by assigning a System Commit Number (SCN) to every transaction. The Oracle Database uses the SCN to ensure consistency, both for data reads (a query includes all committed changes up to SCN <x>), and for recovery with changes being recovered in SCN order.

HVR implements log-based CDC for Oracle by parsing transaction log entries directly. Changes are captured as they get written to the transaction log, but only propagated, in SCN order, after the transaction commit is written to the log. Transactions, or parts of transactions, that are rolled back are discarded. HVR can also replicate DDL changes to tables like create/drop table, add/alter/drop column, and changes to the primary key

Installed on the source server, the HVR executable can read redo and archive logs directly from the file system, including Oracle Automatic Storage Manager (ASM). For maximum flexibility, remote capture through SQL is available, leveraging Oracle's logminer to present raw redo records through an in-memory V$-view. Trigger-based capture is still supported but no longer improved.

Initial time-based positioning is achieved by running queries against V$ views to identify what transaction log file(s) to start with. HVR's capture state file contain file sequence number(s) and relative byte address for re-positioning to ensure no loss of changes. To enable quick capture recovery with long-running transactions in the database HVR checkpoints capture state to disk at an interval so that upon restart the rewind time can be limited to the point in time of the most recent checkpoint rather than the at that point oldest open transaction.

To perform log-based CDC in an Oracle RAC configuration HVR runs on one of the nodes in the cluster, capturing changes for all threads. Upon startup HVR requests the SCAN listener which node to connect to. In case of an instance or node failure, upon restart, CDC will automatically resume where it left off by connecting to a different node.

HVR's log-based CDC maintains transactional consistency as well as data integrity by keeping track of the transaction boundaries and the order of changes within a transaction. In an Oracle RAC environment individual readers per thread present their changes to the coordinator that merges the changes across the threads in SCN order. HVR detects threads going down and coming back up to ensure no data loss.

The impact of log-based CDC is typically well below 10% extra load on a source server (except in a many-node Oracle RAC, with all capture processes running on a single node, the impact may be higher). However to limit impact on the production workload HVR supports log-based CDC from a physical standby database (active or passive), as well as capture running on a separate server that only has access to the archived redo logs (so-called Archive Log Only (ALO) mode).

Target Oracle

HVR uses Oracle’s native connectivity protocol TNS to connect to the database. To apply changes HVR submits SQL statements efficiently through OCI (Oracle Call Interface). Bulk data refreshes, including the initial data load, as well as loads into staging tables for burst integration, are done through efficient direct path append operations.

HVR is an Oracle Gold Partner

ORACLE®
Gold Partner

ABOUT HVR

We accelerate data movement so that you can revolutionize your business. HVR is designed to move large volumes of data FAST and efficiently in modern environments for continuous real-time updates.

Our goal is to keep your data moving and in sync as you adopt new technologies for storing, streaming, and analyzing data. Our scalable solution gives you everything you need for efficient data replication from beginning to end so that you can readily revolutionize your business for the modern world.