Hvrlogrelease

Contents

Name

hvrlogrelease - Manage DBMS logging files when not needed by log-based capture.

Synopsis

hvrlogrelease [*optfile*] [-*options*]

Description

hvrlogrelease works differently depending on database class applied: it either manages log files (for Oracle and Ingres) or frees log file blocks (for SQL Server). Database class is given by one of **-oracle_sid**, **-ingres_db** or **-sqlserver_db** options, which must be provided.

Functionality for Oracle

hvrlogrelease can be used to manage Oracle archive files so that they are available if needed by HVR's log-based capture jobs. The default behavior is that **hvrlogrelease** monitors the 'primary' archive directory where the Oracle archiver process writes these files and makes a copy in a private directory whenever one appears. It then removes these copies when the HVR capture jobs will no longer need their contents.

An alternative behavior is activated by option **-purge**; in this case no copies are made and **hvrlogrelease** simply deletes Oracle archive file from the 'primary' archive directory once it sees these files will no longer be needed by HVR capture jobs. This is useful if archiving was only enabled for HVR's log-based capture and are not needed for backup and recovery.

It relies on 'log release' files that are re-created by log-based capture jobs after each replication cycle. These files are in **$HVR_CONFIG/files** and contain timestamps which allows **hvrlogrelease** to see which archive files could still be needed by HVR replication.

When **hvrlogrelease** is making private copies of archive files (option **-purge** not defined), they are placed in a special directory using a file system 'hard link' to avoid i/o overhead. The path is defined by action **Capture /ArchiveLogPath**. If that is not defined then a default path is derived by adding the suffix **_hvr** to the Oracle archive destination directory.

The files copied by **hvrlogrelease** keep their original 'base name', so parameter **/ArchiveLogFormat** cannot be used.

Command **hvrlogrelease** with option **-purge** can be used with action **Capture** **/ArchiveLogPath /ArchiveLogOnly** to remove files which the capture job no longer needs. In this case, an additional option **-force** must be used.

Functionality for Ingres

hvrlogrelease can be used to manage Ingres journal files so that they are available if needed by HVR log-based capture jobs. The default behavior is that **hvrlogrelease** monitors the directory where the DBMS creates these files, and makes a copy in a private directory whenever one appears. It then removes these copies when the HVR capture jobs will no longer need their contents.

An alternative behavior is activated by option **-purge**; in this case no copies are made and **hvrlogrelease** simply deletes DBMS logging files from the archive or journal directory once it sees these files will no longer be needed by HVR capture jobs. This is only useful if journaling was enabled for HVR's log-based capture and are not needed for backup and recovery.

It relies on 'log release' files that are re-created by log-based capture jobs after each replication cycle. These files are in **$HVR_CONFIG/files** and contain timestamps which allows **hvrlogrelease** to see which DBMS logging files could still be needed by HVR replication.

When **hvrlogrelease** makes private copies of Ingres journal files (option **-purge** not defined) they are placed in special directory using a file system 'hard link' to avoid i/o overhead. The path is defined by environment variable **$HVR_LOG_RELEASE_DIR**. If that is not defined then a default path is derived by adding the suffix **_hvr** to the database's journal directory.

Functionality for SQL Server

hvrlogrelease resets log file blocks from 'taken' into 'free' state. Such blocks can be later reused to prevent excessive log file growth. That log 'truncation' may be needed in the following situations:

- coexistence, when several brands of replication tools need to read same log file;
- multi-capture, when several capture jobs read the same log file;
- in order to reset the bunch of used log file blocks when log has not being truncated for a long period. <br/>

First two scenarios require configuring HVR capture job with automatic log truncation turned off. Then log file needs to be truncated by executing **hvrlogrelease** command from according job ran by schedule or any other means. For more information, see Managing SQL Server Log file truncation.

Notes

- **hvrlogrelease** must be scheduled to run under the DBMS owner's login (e.g. **oracle** or **ingres)** , whereas **hvrmaint** must run under the HVR's login.
- When **hvrlogrelease** is installed on an Oracle RAC, then the **$HVR_CONFIG** directory must be shared between all nodes. Directory **$HVR_TMP** (if configured) should not be shared. Command **hvrlogrelease** should then be scheduled to run on all nodes, but with 'interleaved' timings. For example 0, 20, 40 minutes after each hour on one node and 10, 30, 50 minutes after each hour on the other node.
- Command **hvrlogrelease** does not support archive files located inside Oracle ASM. In this situation the RMAN must be configured to retain the archive files for sufficient time for HVR.

Options

This section describes the options available for command **hvrlogrelease**.

Parameter	Description
-email=*addr1*[;*addr2*]	Email output from **hvrlogrelease** output to *addr1* [and *addr2*]. Requires either option **-smtp_server** or option **-mailer**. Multiple email addresses can be specified, either using a single **-email** option with values separated by a semicolon or using multiple **-email** options.

-email_only_errors	Only send an email if **hvrlogrelease** encounters an error.
-email_activity	Only send an email if **hvrlogrelease** encounters an error or purged or compressed any file.
-email_from=*from*	Specify a *from* address in email header.
-env=*NAME=VALUE*	Set environment variable, such as **$HVR_HOME**. This option can be repeated to set different variables. Values for **$ORACLE_SID** and **$HVR_CONFIG** should be defined with special options **-oracle_sid** and **-hvr_config** respectively.
-help	Provide options list with service information to create UI dialog.
-hvr_config=*dir*	Check for 'log release' files in this **$HVR_CONFIG** directory. This option must be supplied at least once. It can also be supplied several times if multiple HVR installations capture log changes from a single database. **hvrlogrelease** will then purge DBMS logging files only after they have been released by all the HVR installations. If value *dir* contains an asterisk (*) then all matching directories are searched.
-ingres_db=*db*	Only check 'log release' files for Ingres database *db*. If value db contains an asterisk (*) then all Ingres databases are matched. This option can be supplied several times for different databases. Note that **hvrlogrelease** extracts the path **$II_SYSTEM** from matching log release files, so it could affect journal files which are not in the current Ingres installation.
-logrelease_expire=*Nunits*	Instruct command **hvrlogrelease** to ignore any 'log release file' that are too old. Value *units* can be **days**, **hours** or **minutes**. For example, value **4days** could be defined because if the capture job has not run for four days then the replication backlog is so great that a refresh will be needed and the DBMS logging files can be discarded.
-mailer=*cmd*	Mailer command to use for sending emails, instead of sending them via an SMTP server. Requires option **-email**. String **%s** contained in *cmd* is replaced by the email subject and string **%a** is replaced by the intended recipients of the email. The body of the email is piped to *cmd* as **stdin**.
-oracle_sid=*sid*	Only check for 'log release' files for Oracle instance *sid*. If value *sid* contains an asterisk (*) then all Oracle instances are matched. This option can be supplied several times for different instances. Note that **hvrlogrelease** extracts the value of **$ORACLE_HOME** from matching log release files, so it could affect archived redo files which are not in the current Oracle installation.
-output=*fil*	Append **hvrlogrelease** output to file *fil*. If this option is not supplied then output is sent to **stdout**. Output can also be sent to an operator using option **-email**.
-state_dir=*dir*	Create files **hvrlogrelease.pid** and **hvrlogrelease.dirs** in provided directory *dir*. If not supplied these files will be created in directory set in environment variable **$HVR_CONFIG**.
-purge	Purge (i.e. delete) old DBMS logging files (Ingres journals and Oracle archived redo logfiles) and backup files (e.g. Ingres checkpoints) once the 'log release' files indicate that they are no longer needed for any HVR replication. If the 'log release' file is absent or unchanged then nothing is purged, so operators must purge journals/archived redo files manually. This is only useful if journaling/archiving was only enabled for HVR's log-based capture and are not needed for backup and recovery.
-smtp_server=*server*	SMTP server to use when sending email. Value *server* can be either a node name or IP address. Requires option **-email**.

-smtp_user=*user*	Username user for authentication SMTP server if needed.
-smtp_pass=*pass*	Password pass used for authentication on the SMTP server if needed.
-user=*login/pass*	Database authentication login and password.
-sqlserver_db=*db*	Mark SQL Server log file blocks free for database db.
-verbose	Write extra logging to output file.

Deleting Ingres Journals with CKPDB

Command **hvrlogrelease** can encounter problems if Ingres command **ckpdb** is used with option **-d** (delete journal files). This is because Ingres will create new journal files when 'draining' the log file and then delete them so quickly that **hvrlogrelease** does not have time to make a copy. The solution is to create a customized Ingres 'checkpoint template' file which calls **hvrlogrelease** from within **ckpdb**. Two lines of this file need to be changed: those labeled **PSDD** and **PSTD**. The following steps use a Perl expression to change them to call **hvrlogrelease** with specific option file (this file must also be created).

```
$ cd $II_SYSTEM/ingres/files
$ perl -pe 's/(PS[DT]D:\s*)/$1hvrlogrelease \$HVR_CONFIG\/files\
/hvrlogrelease.opt;/' cktmpl.def >cktmpl_hvr.def
$ ingsetenv II_CKTMPL_FILE $II_SYSTEM/ingres/files/cktmpl_hvr.def
```

Examples

Example 1 - manage Oracle archive file copies
The following can be saved in option file **/opt/hvrlogrelease.opt** so that private copies of any Oracle archive files from instance **ORA1020** are available when needed by HVR log-based capture:

```
-env=HVR_HOME=/opt/hvr410/hvr_home
-hvr_config=/opt/hvr410/hvr_config
-oracle_sid=ORA1020
-logrelease_expire=3days
-email_from=hvr@prod.mycorp.com
-email=bob@mycorp.com;jim@mycorp.com
-email_only_errors
```

If Oracle command **rman** is also configured to remove old redo logfiles, then **hvrlogrelease** must be scheduled to run first so that it sees every file before that file is removed by **rman**. This can be done by scheduling both **hvrlogrelease** and **rman** in a single crontab line.

```
00,15,30,45 * * * * su oracle -c "/opt/hvr_home/bin/hvrlogrelease
/opt/hvrlogrelease.opt        >> /tmp/hvrlogrelease.log 2>&1 && rman
@delete_archives >>         /tmp/delete_archives.log 2>&1"
```

Example 2 - manage Ingres journal file copies
The following option file can be used to maintain private copies of any Ingres journal files for database **mydb** so that they are available when needed by HVR log-based capture:

```
-env=HVR_HOME=/opt/hvr410/hvr_home
-hvr_config=/opt/hvr410/hvr_config
-ingres_db=mydb
-email_from=hvr@prod.mycorp.com
-email=bob@mycorp.com;jim@mycorp.com
-email_only_errors
```

Example 3 - mark SQL Server log blocks as free
The following option file can be used to move Log Sequence Number replication pointer for database **mydb** to mark virtual log file blocks as free:

```
-sqlserver_db=mydb
-user=my_login/my_pwd
-output=c:\hvr\hvr_config\log\hvrlogrelease-hub-mydb.out
```

Files

- **HVR_CONFIG**
  - **files**
    - **hvrlogrelease.dir** Cache of search directories.
    - **hvrlogrelease.pid** Process ID of current **hvrlogrelease** command.
    - [*node*-]*hub-chn-loc***.logrelease** Log release file, containing the time. Recreated by each log-based capture job cycle and used by **hvrlogrelease**.