Deep Dive into DBFS

Christian Gohmann

HrOUG Conference 2022

trivadis
Part of Accenture

HALLO, GRÜEZI, HI!

CHRISTIAN GOHMANN

- Principal Consultant, Instructor and Tool Owner of db*BACKUP
- Working with Oracle since 2006
- Focused on High Availability Solutions, Migration Projects, Backup & Recovery and Cloud Technologies
- Oracle ACE Associate

1994
FOUNDATION

>2200
TRAINING PARTICIPANTS PER YEAR

2100
ORDERS PER YEAR

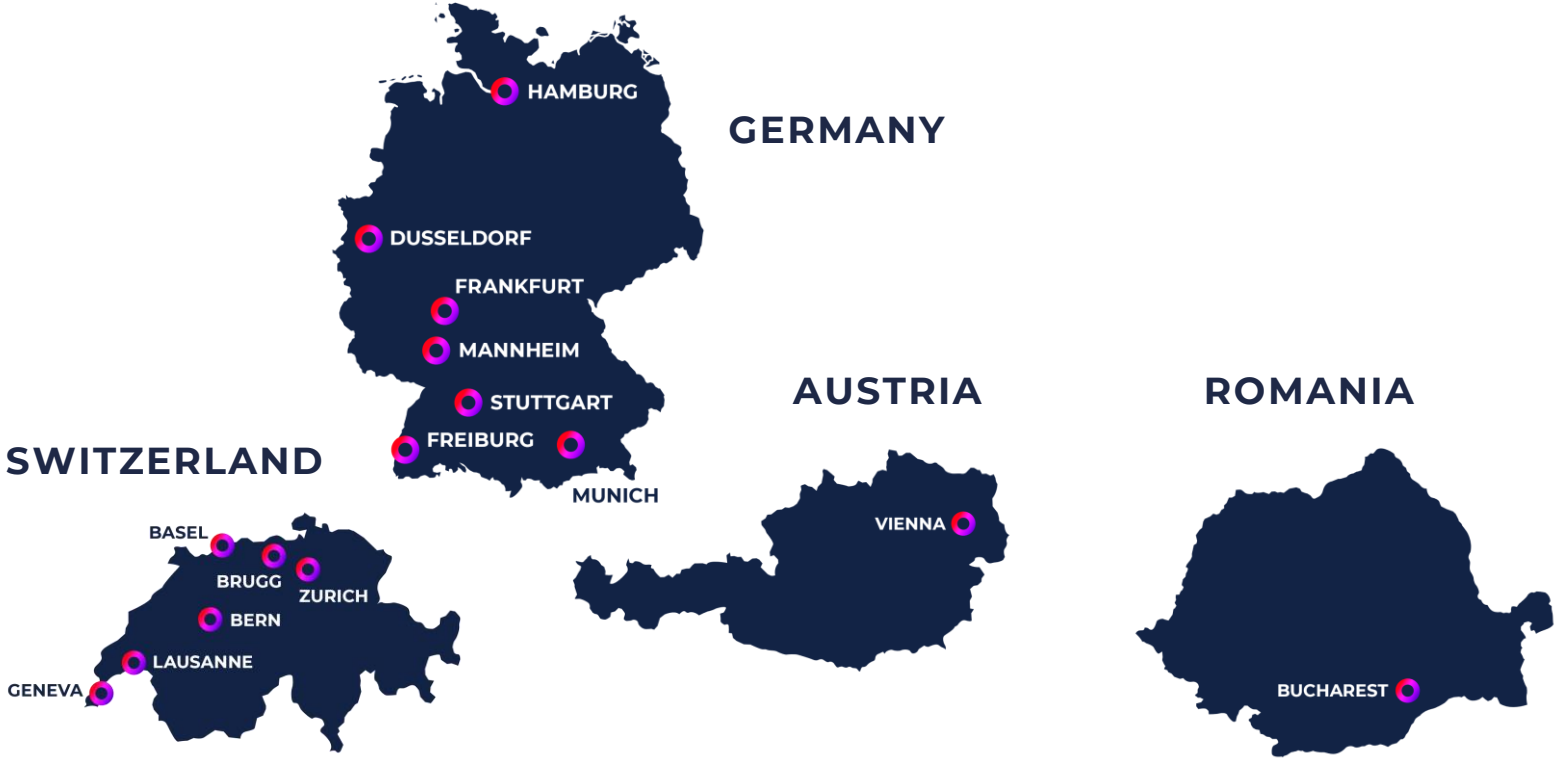
>750
CUSTOMERS PER YEAR

123 Mio.
TURNOVER

>300
ACTIVE SLAs

trivadis
Part of Accenture

4 OUR WORKSPACES



trivadis
Part of Accenture

AGENDA

1. Introduction
2. DBFS
3. ACFS
4. Comparison

INTRODUCTION

7 CHALLENGE

We need to store structured data AND unstructured (binary) data!

- In-place or out-of-place storing of binary files?
- The size of the database will increase, do we except issues because of this?
- What about backup and restore of (single) binary files and the database?
- How can we easily access the files – our application supports only file system access?
- …

8 LARGE OBJECT (LOB) DATA TYPES

Data Type	Maximum Size	Usage
BFILE		Pointer to a binary file outside the database
BLOB	4 GB	Unstructured binary data
CLOB	4 GB	Character data
NCLOB	4 GB	Character data, **national character set is used**
LONG	2 GB	Character data
LONG RAW	2 GB	Unstructured binary data
RAW	2.000 Bytes	Unstructured binary data

LONG, LONG RAW and RAW data types should not be used nowadays.

9 HANDLING OF LOBS 1/2

- Working with LOBs is sometimes not very straight forward and requires **additional PL/SQL code**
- PL/SQL package DBMS_LOB is used to work with LOBs
- To access files outside the database a DIRECTORY object is required

```
SQL> CREATE OR REPLACE DIRECTORY FILE_DIR AS '/data/files';
SQL> GRANT READ, WRITE, EXECUTE ON DIRECTORY FILE_DIR TO SCOTT;
```

User or role

Use a least privilege approach for the access and the permissions (READ, WRITE, EXECUTE) to a DIRECTORY object.

10 HANDLING OF LOBS 2/2

- Example: Loading of a PDF file into a table

DIRECTORY object

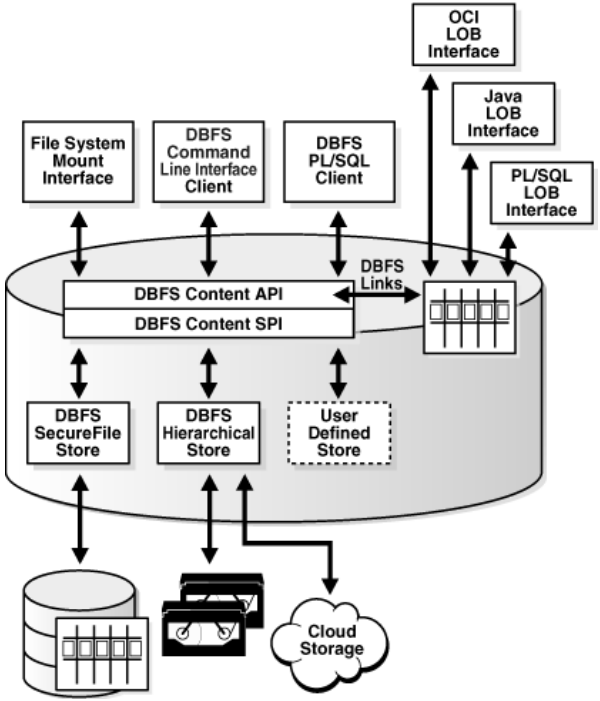
```
SQL> DECLARE
   v_file         BFILE := BFILENAME('FILE_DIR', 'report_2022.pdf');
   v_file_content BLOB;
   v_src_offset   NUMBER := 1;
   v_dest_offset  NUMBER := 1;
BEGIN
   DBMS_LOB.CREATETEMPORARY(lob_loc => v_file_content, cache => TRUE);

   DBMS_LOB.FILEOPEN(file_loc => v_file, open_mode => DBMS_LOB.FILE_READONLY);
   DBMS_LOB.LOADBLOBFROMFILE(dest_lob => v_file_content, src_bfile => v_file,
      amount => DBMS_LOB.GETLENGTH(v_file), dest_offset => v_dest_offset,
      src_offset => v_src_offset);
   DBMS_LOB.FILECLOSE(file_loc => v_file);

   INSERT INTO REPORT_FILES (name, report) VALUES('Report 2022', v_file_content);
END;
/
```

DATABASE FILESYSTEM (DBFS)

12 OVERVIEW

- Part of the database since 11g Release 2
- Allows the creation of **local file systems within the database** to access files and directories stored in tables
  - LOBs are used to store the objects
- Contains a **server and client component**
  - The Oracle database is the server
  - Similarities to NFS
- DBFS can be mounted on the operating system level to access the files without changing the application

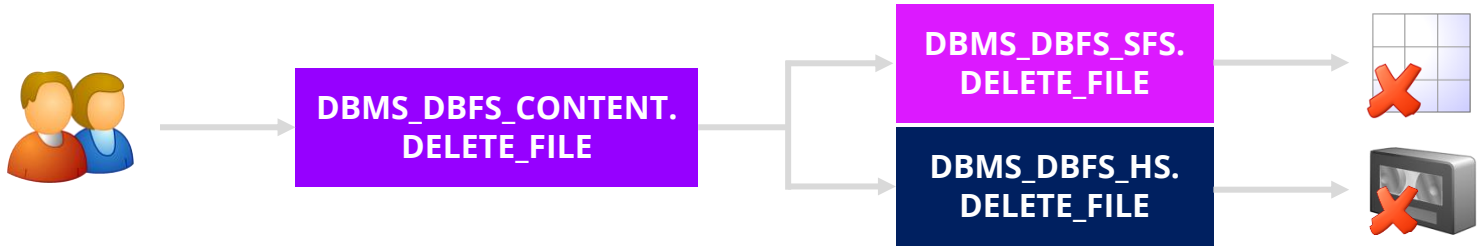
13 ARCHITECTURE



- **Source:** https://docs.oracle.com/en/database/oracle/oracle-database/21/adlob/what-is-database-file-system.html#GUID-364B5298-6E85-4C02-A6BF-CD895FAE207A

14 DBFS CONTENT API

- **Client-side programmatic API**
  - PL/SQL package DBMS_DBFS_CONTENT
- **Collection of functions and procedures** that provides a file system-like abstraction
- Supports different DBS Content Store Providers
  - Dynamic SQL is internally used which can cause runtime errors
- Example: Deletion of a file



The package is created with the script $ORACLE_HOME/rdbms/admin/dbmscapi.sql.

15 DBFS CONTENT STORE PROVIDER INTERFACE (SPI)

- Collection of required **functions and procedures specifications**
  - Each DBFS Content Store needs to implement this interface
  - PL/SQL package **DBMS_DBFS_CONTENT_SPI** has no body

```
SQL> SELECT object_name, object_type FROM dba_objects
      WHERE object_name = 'DBMS_DBFS_CONTENT_SPI';

OBJECT_NAME                    OBJECT_TYPE
------------------------------ -----------------------
DBMS_DBFS_CONTENT_SPI          PACKAGE
```

- Responds to calls of the DBFS Content API

The specification is created with the script $ORACLE_HOME/rdbms/admin/dbmscapi.sql.

CONTENT STORE PROVIDERS

DBFS Secure Files Store (DBMS_DBFS_SFS)

Focus of this presentation

- A table with LOB column is used to store the file system data
- POSIX-like file system capabilities (including locking since 12c Release 2)

DBFS Hierarchical Store (DBMS_DBFS_HS)

- Files are stored on any tape storage that is supported by RMAN or on cloud storage systems (e.g., Amazon S3) - https://tinyurl.com/2tsbk29t
- Only metadata is written to the database

User-defined Store

- Allows the creation of a user-defined store within the database
- PL/SQL or Java can be used
- No operating system code is required
- Example: https://tinyurl.com/2p9b4v7s

17 REQUIRED PERMISSIONS

- File systems are created in tables in the own schema
- To create and access a DBFS, the role **DBFS_ROLE** is required
  - Contains only object privileges

```
SQL> GRANT DBFS_ROLE TO SCOTT;
SQL> GRANT CREATE TABLE TO SCOTT;
```

- If features like shrinking and reorganization will be used, additional grants are required

DBFS reorganization uses DBMS_REDEFINITION internally.

18 CREATE FILE SYSTEM 1/2

- **File system names are case sensitive and user-unique**
- Additional licenses are required to use advanced features like compression, deduplication, encryption or partitioning

```
SQL> BEGIN
   DBMS_DBFS_SFS.CREATEFILESYSTEM (
      store_name => 'MYFS1',
      tbl_tbs => NULL, tbl_name => 'REPORT_FILES', use_bf => FALSE,
      do_dedup => TRUE,
      do_compress => TRUE, compression => DBMS_DBFS_SFS.COMPRESSION_MEDIUM,
      do_encrypt => TRUE, encryption => DBMS_DBFS_SFS.ENCRYPTION_AES256,
      do_partition => TRUE, partition_key => DBMS_DBFS_SFS.PARTITION_BY_ITEM
   );
END;
/
```

DBFS Secure Files Store is used

If set to TRUE, Basic Files are used

Before and after the execution of this procedure, a commit is executed – like a DDL.

19 CREATE FILE SYSTEM 2/2

- Use the DBMS_DBFS_SFS functions to retrieve details about existing file systems

```
SQL> SQL> SELECT schema_name, store_name, table_name, created
      FROM TABLE(DBMS_DBFS_SFS.LISTFILESYSTEMS);

SCHEMA_NAME     STORE_NAME           TABLE_NAME      CREATED
--------------- -------------------- --------------- ----------------------------
SCOTT           MYFS1                REPORT_FILES    27.05.22 09:08:35,759469

SQL> SELECT schema_name, table_name, created FROM TABLE(DBMS_DBFS_SFS.LISTTABLES);

SCHEMA_NAME     TABLE_NAME      CREATED
--------------- --------------- ------------------------------
SCOTT           REPORT_FILES    27.05.22 09:08:35,759469
```

If no table name was provided, a system-generated name (SFS$_FST_nn) is used

trivadis
Part of Accenture

20 ASSOCIATE STORE PROVIDER

- Associate the newly created store with a provider

```
SQL> BEGIN
   DBMS_DBFS_CONTENT.REGISTERSTORE (
      store_name => 'MYFS1',
      provider_name => 'DBFS Secure Files Store',
      provider_package => 'DBMS_DBFS_SFS'
   );
END;
/

SQL> SELECT store_name, provider_name, provider_pkg, store_features
       FROM TABLE(DBMS_DBFS_CONTENT.LISTSTORES);

STORE_NAME      PROVIDER_NAME                    PROVIDER_PKG    STORE_FEATURES
--------------- -------------------------------- --------------- --------------
MYFS1           DBFS SecureFiles Store           DBMS_DBFS_SFS         79822999
```

Seems to be ignored, can be anything

Bit mask of supported features

trivadis
Part of Accenture

21 CHECK FEATURES OF THE STORE

- Converts the bit mask of the store features into a human readable output

```
SQL> SELECT *
       FROM TABLE(DBMS_DBFS_CONTENT.DECODEFEATURES(
                  DBMS_DBFS_CONTENT.GETFEATURESBYNAME('MYFS1')));

FEATURE_NAME                    FEATURE_MASK FEA
------------------------------- ------------ ---
folders                                    1 YES
foiat                                      2 YES
nowait                                     4 YES
acls                                       8 NO
links                                     16 YES
link_deref                                32 NO
references                                64 NO
locking                                  128 YES
…
getattr_view                        67108864 YES
```

22 MOUNT THE STORE

- To access a store from a client, it needs to be mounted

```
SQL> BEGIN
   DBMS_DBFS_CONTENT.MOUNTSTORE (
      store_name => 'MYFS1',
      store_mount => 'myfs1_mnt',
      read_only => FALSE
   );
END;
/

SQL> SELECT store_name, store_mount, mount_properties
       FROM TABLE(DBMS_DBFS_CONTENT.LISTMOUNTS);

STORE_NAME      STORE_MOUNT     MOUNT_PROPERTIES(PROPNAME, PROPVALUE, TYPECODE)
--------------- --------------- ------------------------------------------------
MYFS1           myfs1_mnt       DBMS_DBFS_CONTENT_PROPERTIES_T(DBMS_DBFS_CONTENT_P
                                ROPERTY_T('owner', NULL, 9),…)
```

The internal DBFS mount point

23 DBFS CREATION VIA SQL SCRIPT

- Instead of running all required steps manually, the scripts **dbfs_create_filesystem.sql** or **dbfs_create_filesystem_advanced.sql** can be used
  - As store provider, the DBFS Secure Files Store is used

```
SQL> @?/rdbms/admin/dbfs_create_filesystem.sql USERS MYFS2

-- Allows the configuration of compression, deduplication, encryption and
-- partitioning
SQL> @?/rdbms/admin/dbfs_create_filesystem_advanced.sql USERS MYFS2
     compress-medium deduplicate noencrypt partition-by-itemname
```

The name of the store mount point is equal to the file system name.

CLIENT ACCESS METHODS 1/4

PL/SQL Client Interface

- Use DBMS_DBFS_CONTENT to interact with the contents of a store
- Allows transactions and read consistency

```
DECLARE
   v_file      BLOB;
   v_content   VARCHAR2(1000)  := 'Hello World!';
   v_props     DBMS_DBFS_CONTENT.PROPERTIES_T;
BEGIN
   …
   -- Create directory
   DBMS_DBFS_CONTENT.CREATEDIRECTORY('/myfs1_mnt/Data', v_props);

   -- Create file
   DBMS_DBFS_CONTENT.CREATEFILE('/myfs1_mnt/Data/hello_world.txt', v_props, v_file);

   -- Write content to file
   DBMS_LOB.writeappend(v_file, LENGTH(v_content), UTL_RAW.CAST_TO_RAW(v_content));
   COMMIT;
END;
/
```

25 CLIENT ACCESS METHODS 2/4

DBFS Client Command-line Interface

- Binary dbfs_client is shipped with a full Oracle client or the database software
- Implements simple file system commands, like cp, mkdir, ls, rm

```
$> dbfs_client scott@MYDB --command ls –a dbfs:/myfs1_mnt
Password:
drwxr-xr-x       root       root            0    May 28 22:01    dbfs:/myfs1_mnt

$> dbfs_client scott@MYDB --command cp /files/rep_2022.pdf dbfs:/myfs1_mnt
Password:
files/rep_2022.pdf -> dbfs:/myfs1_mnt/rep_2022.pdf
```

Store mount

- Direct connection to the database
- No need to mount the file system on the host machine

Use a wallet to avoid the interactive password prompt.

CLIENT ACCESS METHODS 3/4

File System Mount Interface

- On Linux and Solaris, dbfs_client can be used to create a file system mount point
  - Filesystem in User Space (FUSE) library is used

```
$> yum install fuse
$> ln –s /lib64/libfuse.so.2 /lib64/libfuse.so
$> ldconfig
```

MOS note 2595545.1

- Transparent access to the database
- Standard file system calls are translated into OCI calls

```
$> dbfs_client –o wallet /@MYDB –o mfs_mount=/mnt/dbfs
$> ls –l /mnt/dbfs/myfs1_mnt/
-rw-r--r-- 1 root root 218 May 28 22:56 rep_2022.pdf
```

A little bit slower compared to direct access with dbfs_client because of the FUSE layer.

CLIENT ACCESS METHODS 4/4

DBFS Links

- Secure Files LOB locaters are used to store files outside of the database
  - E.g., tape systems, cloud or different file systems
- The file is copied to a local area when a write operation is performed
- Reads can use the local area or a streaming mechanism
- Deletes only delete the link, the file behind it will stay untouched

```
SQL> BEGIN
   DBMS_LOB.MOVE_TO_DBFS_LINK(
      lob_loc => <BLOB or CLOB>,
      storage_path => <New location>
   );
END;
/
```

28 READ-ONLY VIEWS

- Contents of a file system can be obtained by using **read-only views**

```
SQL> SELECT pathname, pathtype, filedata FROM DBFS_CONTENT WHERE store = 'MYFS1';

PATHNAME                       PATHTYPE        FILEDATA
------------------------------ --------------- ------------------------------
/myfs1_mnt                     directory
/myfs1_mnt/rep_2022-pdf        file            3132372E302…E6C6F63616C646F6D61
/myfs1_mnt/.sfs/content        directory
…
/myfs1_mnt/.sfs                directory

SQL> SELECT pathname, property_name, property_value
       FROM DBFS_CONTENT_PROPERTIES WHERE pathname LIKE '%rep_2022.pdf'

PATHNAME                       PROPERTY_NAME                   PROPERTY_VALUE
------------------------------ ------------------------------- ----------------
/myfs1_mnt/rep_2022.pdf        posix:blksize                   8060
/myfs1_mnt/rep_2022.pdf        posix:gid                       0
…
```

ASM CLUSTER FILESYSTEM (ACFS)

INTRODUCTION

- Introduced with Oracle 11g Release 2
- **General purpose file system** based on Automatic Storage Management (ASM)
  - Used to store application files outside of a database
  - Starting with 12c Release 1, all database files can be placed on ACFS
- Part of the **Grid Infrastructure installation**
  - Oracle Restart and Cluster configurations are supported
- Provides advanced features like encryption, snapshots, replication, and many more
- An ACFS can accessed through an **operating system mount point**
- Check MOS note ACFS Support On OS Platforms (Certification Matrix). (Doc ID 1369107.1) to see which platforms are supported

Some features, like for example ACFS Replication are only available in a cluster configuration.

31 CREATE AN ACFS 1/2

- Different ways to create a new ACFS
  - ASM Configuration Assistant (ASMCA)
  - ASM Command-line Tool (ASMCMD)
  - Enterprise Manager Cloud Control
- First create an **ADVM volume** in an existing ASM diskgroup

```
ASMCMD> volcreate -G ACFS_DATA -s 1G appdatavol1
ASMCMD> volinfo -G ACFS_DATA -a
Diskgroup Name: ACFS_DATA

         Volume Name: APPDATAVOL1
         Volume Device: /dev/asm/appdatavol1-157
         State: ENABLED
         Size (MB): 1024
         …
         Usage:
         Mountpath:
```

Save it for further steps

32 CREATE AN ACFS 2/2

- Create the file system using **mkfs.acfs**

```
$> mkfs.acfs /dev/asm/appdatavol1-157
mkfs.acfs: version                   = 21.0.0.0.0
mkfs.acfs: on-disk version           = 50.0
mkfs.acfs: volume                    = /dev/asm/appdatavol1-157
mkfs.acfs: volume size               = 1073741824  (   1.00 GB )
mkfs.acfs: file system size          = 1073741824  (   1.00 GB )
mkfs.acfs: Format complete.
```

- Register the file system as Grid Infrastructure resource (only with Oracle Clusterware)
  - On Oracle Restart mount it manually using the mount command

```
$> srvctl add filesystem -device /dev/asm/appdatavol1-157 -path /app_data \
     -user oracle –fstype ACFS –autostart ALWAYS

$> srvctl start filesystem -device /dev/asm/appdatavol1-157
```

Operating system mount point

trivadis
Part of Accenture

COMPARISON

BACKUP AND RECOVERY

DBFS

- RMAN can be used to backup the database, including all DBFS'
- A recovery with RMAN would result in a consistent state of database and files
- An additional file system backup can be created to easily restore single files
    - Afterwards the file is added to the DBFS

ACFS

- Snapshots can be created and used during a restore
- File system backups can be created
- Point-in-time recovery is not possible (or only in a limited way)

RECPLICATION

DBFS

- Oracle Data Guard can be used to replicate the whole database to a different server
- DBFS Links need to be handled differently
  - E.g., file system sync

ACFS

- ACFS replication can be configured between two servers
  - Only available for cluster configurations
- Replication can be filtered using ACFS Tagging

36 PERFORMANCE 1/3

- **500 files (each 10 KiB)** were copied from a local file system to the target file system
- For DBFS a wallet was used to avoid the interactive password prompt
- **A file-by-file copy was performed**

File System	Interface	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8	# 9	# 10	Avg	Med
ACFS		1,7	1,1	1,2	1	0,9	1	0,9	1	1,1	1	**1,09**	**1**
DBFS	Command-line Client	618,8	618,0	603,2	598,7	627,4	625,8	616,8	615,8	586,7	602,9	**611,5**	**616,3**
	Mount Interface	13,5	14,2	14,5	14,3	14,0	13,7	13,8	13,4	13,5	13,4	**13,8**	**13,8**

37 PERFORMANCE 2/3

- **500 files (each 10 KiB)** were copied from a local file system to the target file system
- For DBFS a wallet was used to avoid the interactive password prompt
- **A bulk copy was performed**

File System	Interface	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8	# 9	# 10	Avg	Med
ACFS		0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,0	0,1	**0,1**	**0,1**
DBFS	Command-line Client	7,5	7,6	7,0	7,4	8,3	7,3	7,4	7,2	8,0	7,6	**7,5**	**7,5**
	Mount Interface	13,1	13,2	12,9	14,3	14,2	15,2	13,9	14,9	13,2	13,6	**13,9**	**13,8**

38 PERFORMANCE 3/3

- **One 1 GB file** was copied from a local file system to the target file system
- For DBFS a wallet was used to avoid the interactive password prompt

File System	Interface	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8	# 9	# 10	Avg	Med
ACFS		1,4	1,6	1,8	1,3	1,3	1,8	1,2	1,8	1,1	1,8	**1,5**	**1,5**
DBFS	Command-line Client	13,6	16,0	18,4	17,4	13,7	12,9	13,5	17,6	13,8	12,8	**15,0**	**13,8**
	Mount Interface	61,5	56,3	63,1	55,5	61,0	59,4	54,1	56,1	60,2	55,2	**58,2**	**57,9**

FURTHER INFORMATION

40 LINKS

- **Oracle SecureFiles and Large Objects Developer's Guide 21c**
  https://docs.oracle.com/en/database/oracle/oracle-database/21/adlob/introducing-database-file-system.html#GUID-B7A83817-F0D6-4A09-AE98-DFC966783109
- **Oracle PL/SQL Packages and Types Reference 21c**
- https://docs.oracle.com/en/database/oracle/oracle-database/21/arpls/DBMS_DBFS_CONTENT.html#GUID-EB25764A-3FC0-4114-B8C0-7A85E2FA3FE1
- **Oracle ASM Cluster File System Administrator's Guide 21c**
  https://docs.oracle.com/en/database/oracle/oracle-database/21/acfsg/
- **My Oracle Support**
  https://support.oracle.com

41 QUESTIONS AND ANSWERS

CHRISTIAN GOHMANN

Principal Consultant

+49 211 58 6664 702
christian.gohmann@trivadis.com
https://www.linkedin.com/in/christian-gohmann/
@CGohmannDE
https://www.christian-gohmann.de