

Daniel Hillinger

Database Administrator

VALUE
TRANSFORMATION
SERVICES
an IBM subsidiary

Comparison of ACFS and DBFS

Index **Exadata** Backup Clone

Solaris Performance Tuning Patching TAF

DBA **Grid Infrastructure** ONS Database Vault

Redhat NFS Partitioning **Clusterware** Client Linux

ZFS **RAC** FAN Security Dataguard ACFS

Disaster Recovery **ASM** Replication

High Availability Restore DBFS

Founded 1. September 2013

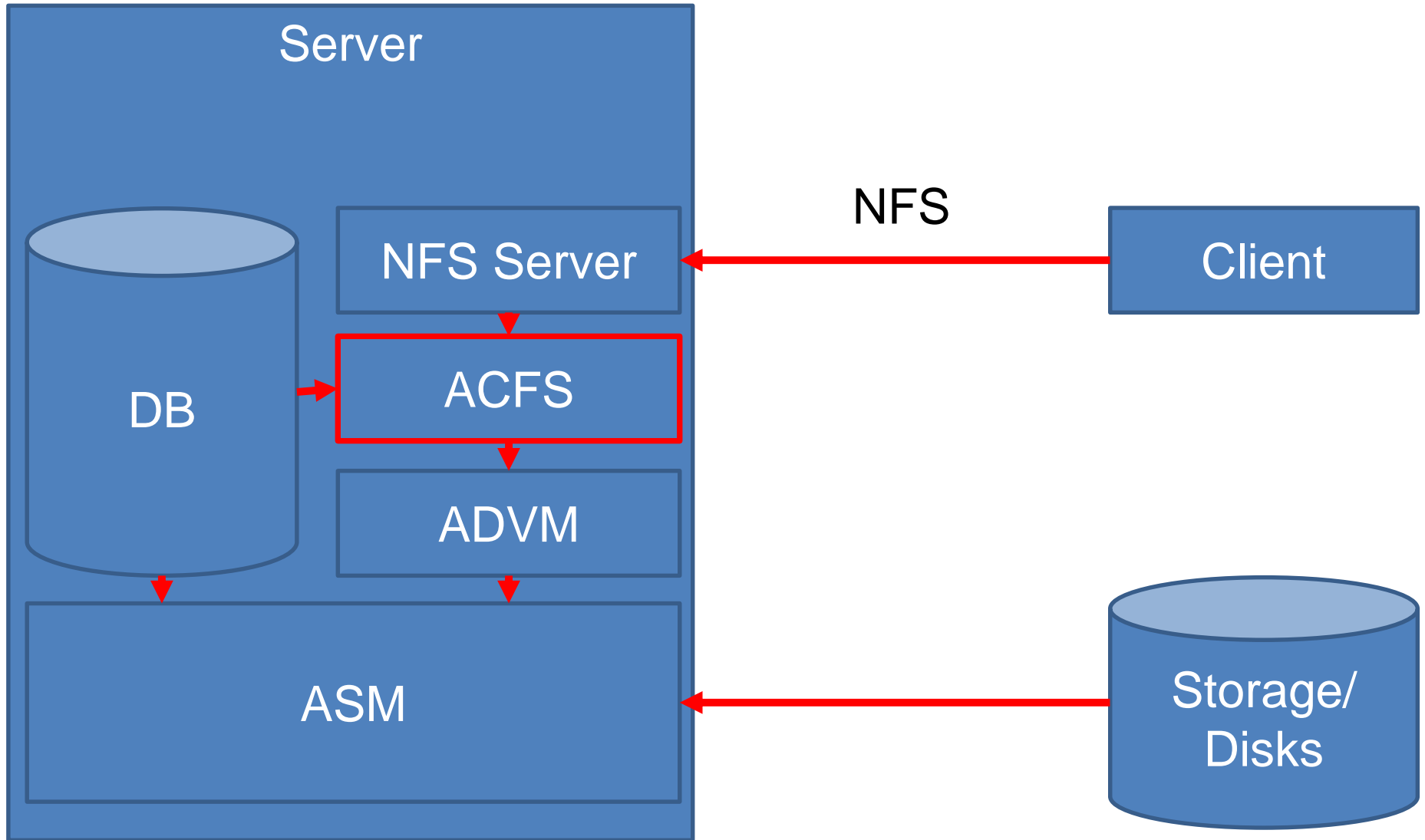
Joint Venture between IBM and Unicredit

- Over 1000 Employees
- 6 Countries
- 6 Datacenters

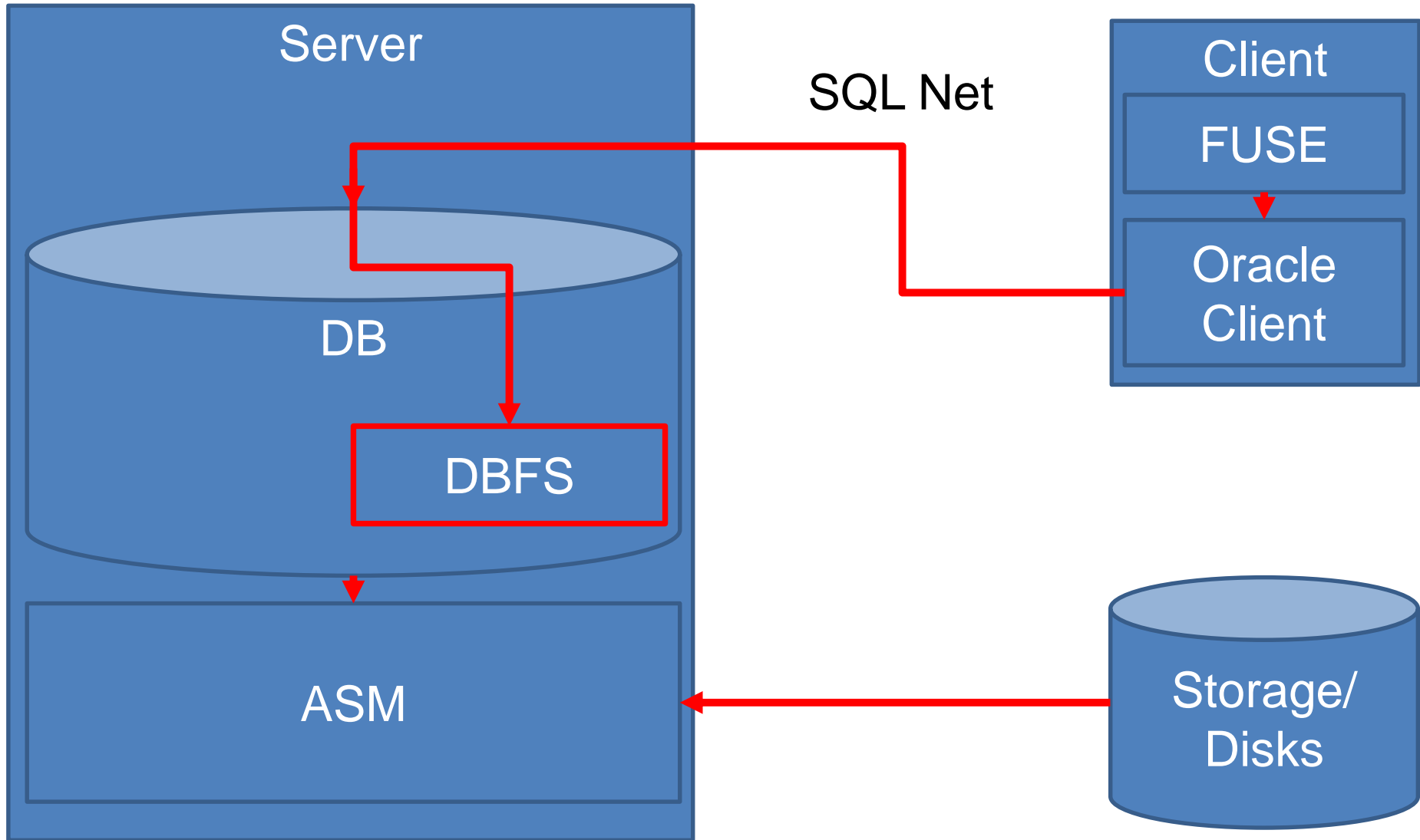
www.v-tservices.com

- Architectural Overview
- Presentation to the Client
- Presentation to the Server/DB
- High Availability and Disaster Recovery
- Extended Architecture
- Setup / Installation
- Tests
- Use cases
- Perspective 12c

Architectural Overview - ACFS



Architectural Overview - DBFS



- Standard NFS mount

```
$ mount
```

```
<IP>:/acfs on /acfs type nfs
```

```
(rw,bg,hard,nointr,rsize=32768,wsiz=32768,tcp,actimeo=0,nfsvers  
=3,timeo=600,addr=<IP>)
```

```
$ df -hTt nfs
```

```
<IP>:/acfs  nfs      170G   16G   154G   10% /acfs
```

- DBFS is based on FUSE (Filesystem in USErspace)
- Oracle Client required

```
$ mount
```

```
dbfs-@DBFS_DBFS:/ on /dbfs type fuse  
(rw,nosuid,nodev,max_read=1048576,default_permissions,allow_oth  
er)
```

```
$ df -hTt fuse
```

```
dbfs-@DBFS_DBFS:/ fuse 4.5G 313M 4.2G 7% /dbfs
```

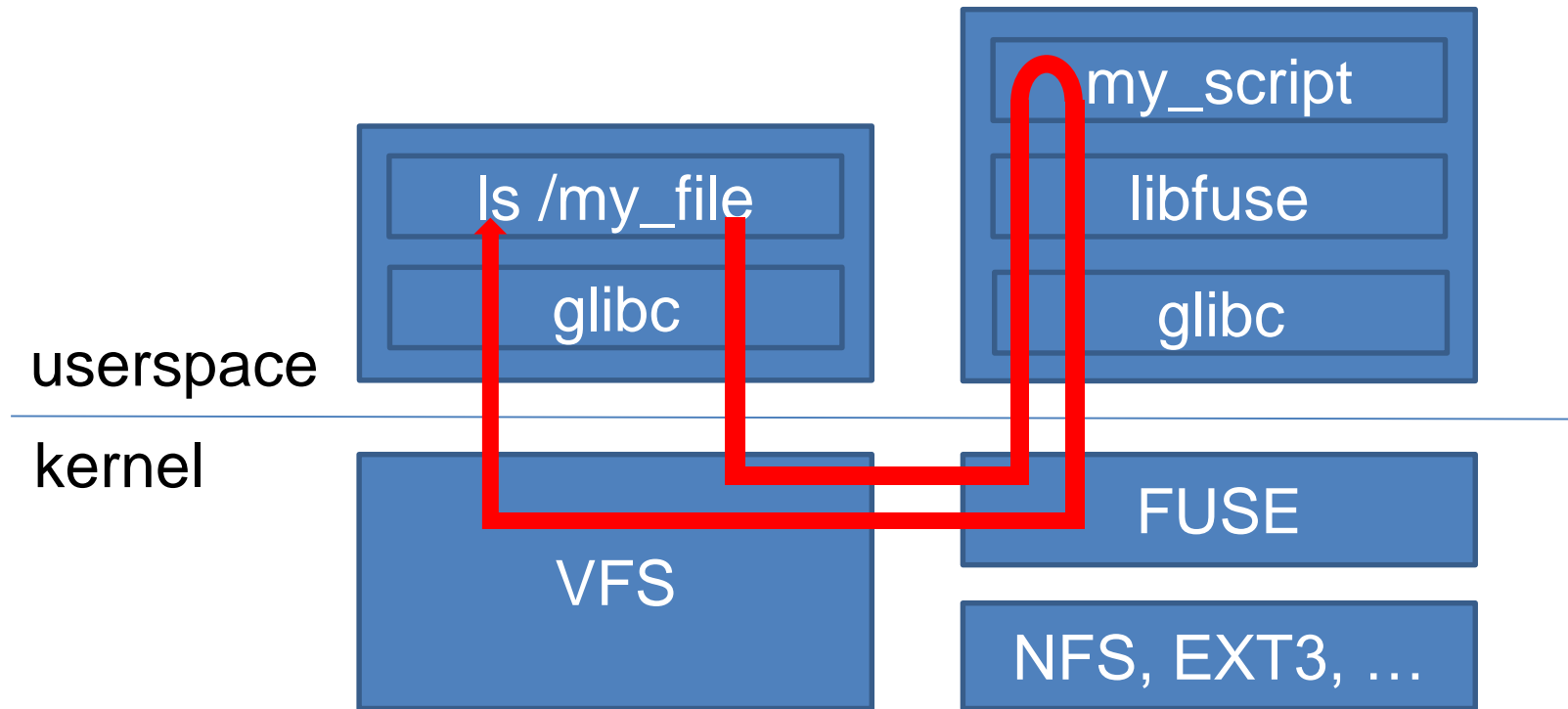
Restrictions:

ioctl, locking, asynchronous I/O, O_DIRECT file opens, hard links, pipes, and other special file modes



Filesystem in UserSpace

- Usable by non privileged users
- Available on Linux and Solaris 11



ASM Dynamic Volume Manager (ADVM) provides a block device in /dev/asm Mounted as standard Linux filesystem

```
$ df -hTt acfs
```

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
/dev/asm/lv_acfs-46	acfs	170G	16G	154G	10%	/acfs

```
$ lsmod
```

oracleacfs	2001119	3	
oracleadv	250040	7	
oracleoks	427736	2	oracleacfs,oracleadv

Can be accessed with

- External Tables
- UTL_FILE
- Datapump
- ...

Server:

Not visible for the server

But could be configured as client

DB:

Access via SQL

```
SELECT UTL_RAW.cast_to_varchar2(filedata) AS filedata
FROM   dbfs_content
WHERE  path = '/dbfs/test1';
```

High Availability(HA):

automated solution in order to conceal the failure of a component

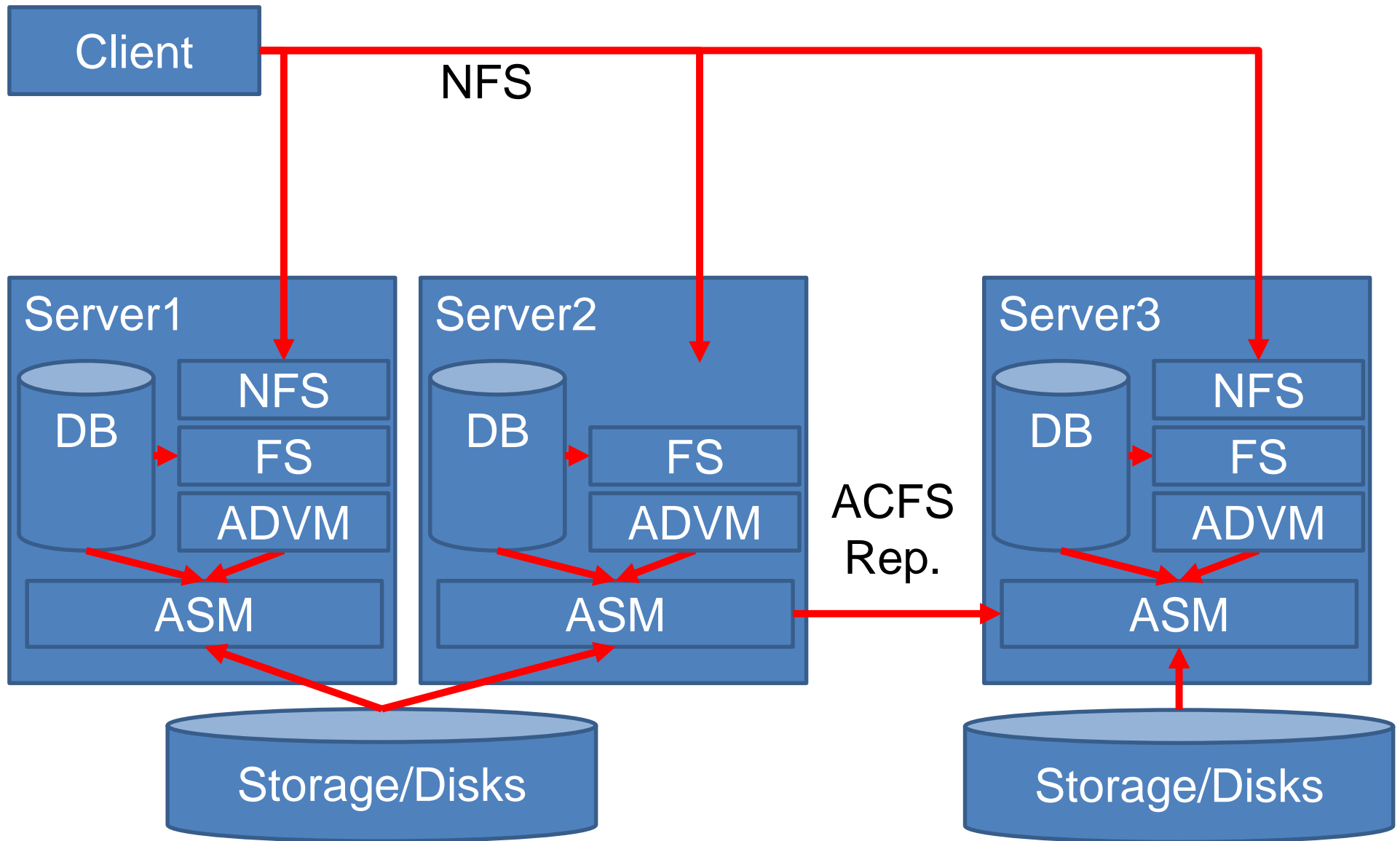
Disaster Recovery(DR):

steps to restore a minimum operation after a major failure, usually manually

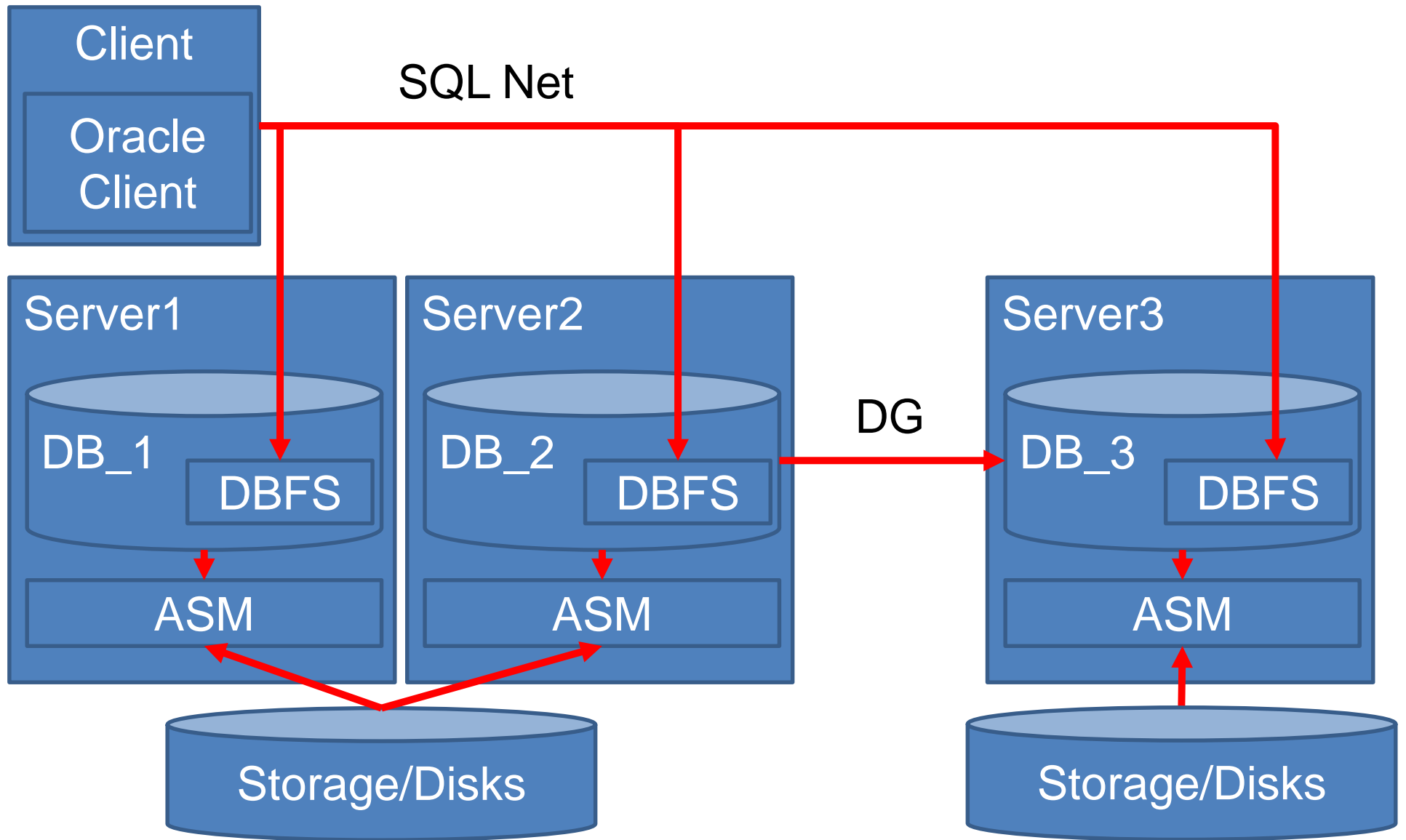
High Availability and Disaster Recovery

	ACFS	DBFS
High Availability		
without interruption	None	RAC
with interruption	Failover NFS Server	Failover DB
Desaster Recovery		
without dataloss	None	Dataguard
with dataloss	ACFS Replication/Rsync	Rsync
Backup	Filesystem	RMAN/Filesystem

Extended Architecture - ACFS



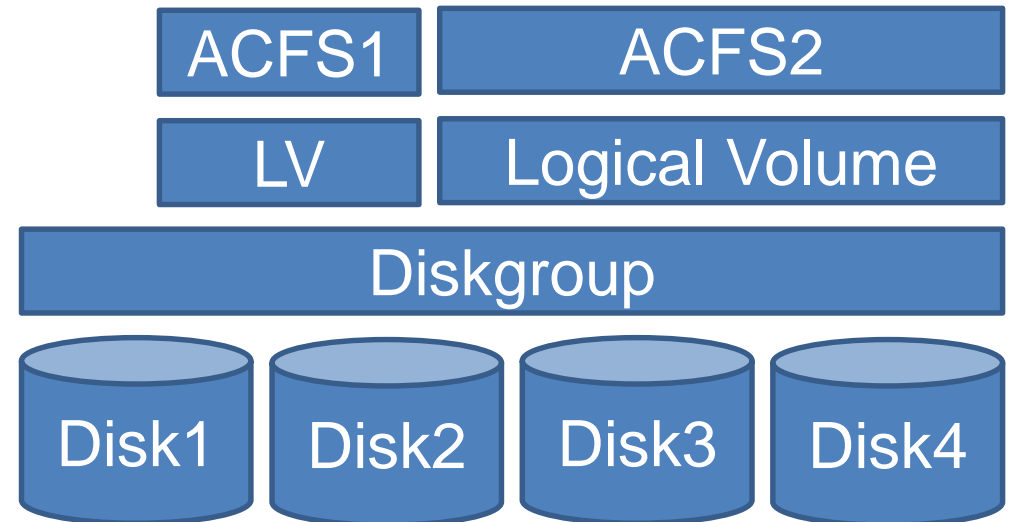
Extended Architecture - DBFS



Supported on Linux, Solaris, AIX and Windows

Requires ASM

Diskgroup can be shared



Dedicated Tablespace

Dedicated user with create session, table, view, procedure privileges and granted dbfs_role

Filesystem creation with script:

```
$ORACLE_HOME/rdbms/admin/dbfs_create_filesystem.sql
```

Files are stored as LOBs in a table called as the filesystem name

Setup / Installation - DBFS

```
SQL> desc mydbfs
```

Name	Null?	Type
.....	-----	-----
...		
PATHNAME	NOT NULL	VARCHAR2 (1024)
FILEDATA		BLOB
POSIX_MODE		NUMBER (38)
POSIX_UID		NUMBER (38)
POSIX_GID		NUMBER (38)
...		

1. Shutdown server hosting NFS server
NFS server switched to another node
2. Kill NFS server
NFS server was restarted
3. Umount ACFS
Umount hanging – not possible

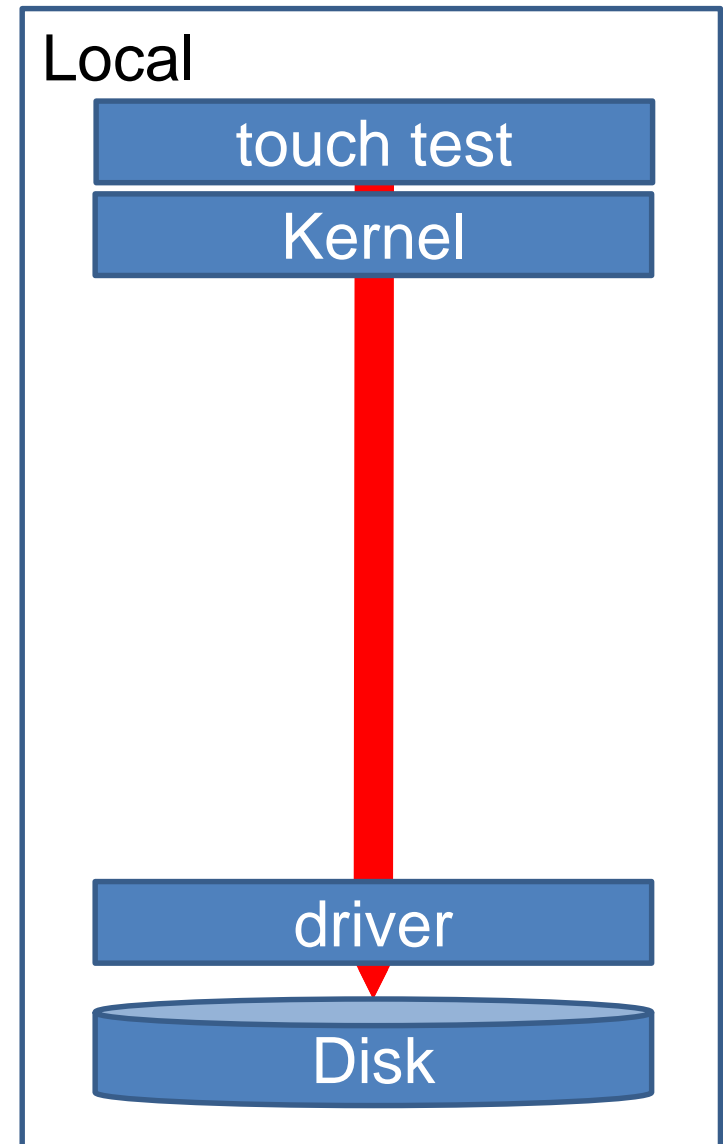
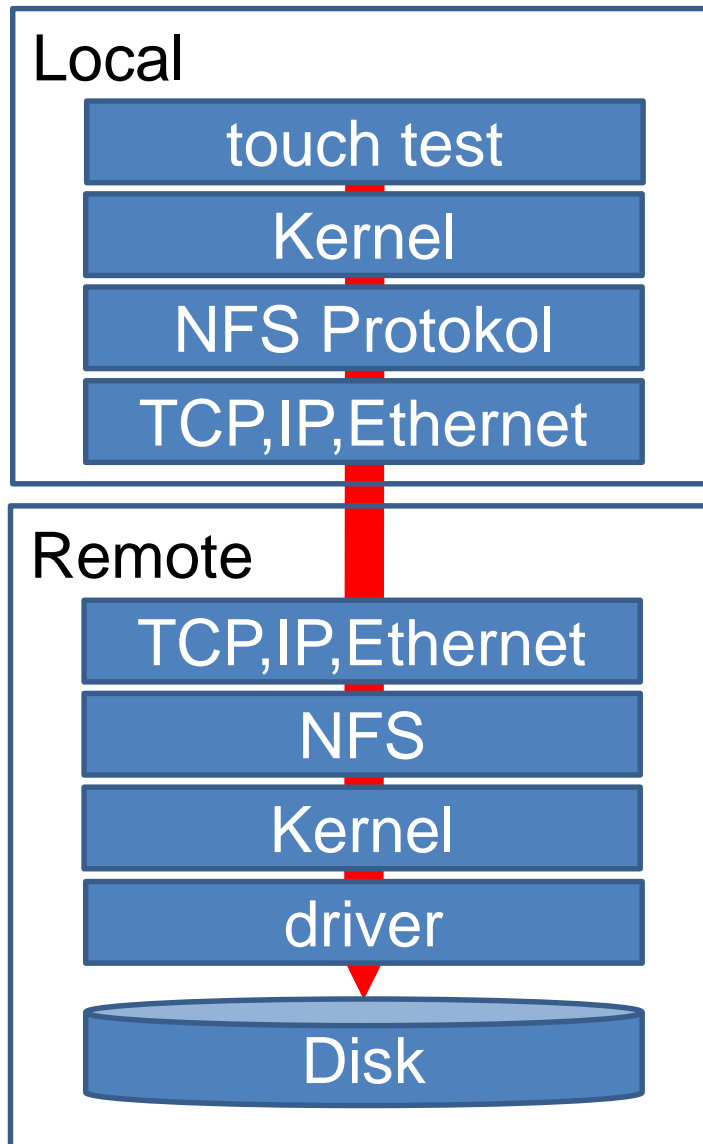
The FS was always available for the client!

1. Shutdown one server
session failover to other instance
2. Kill one instance
session failover to other instance
3. Kill listener
sessions are established only during the mount
4. Client: Kill dbfs_client process

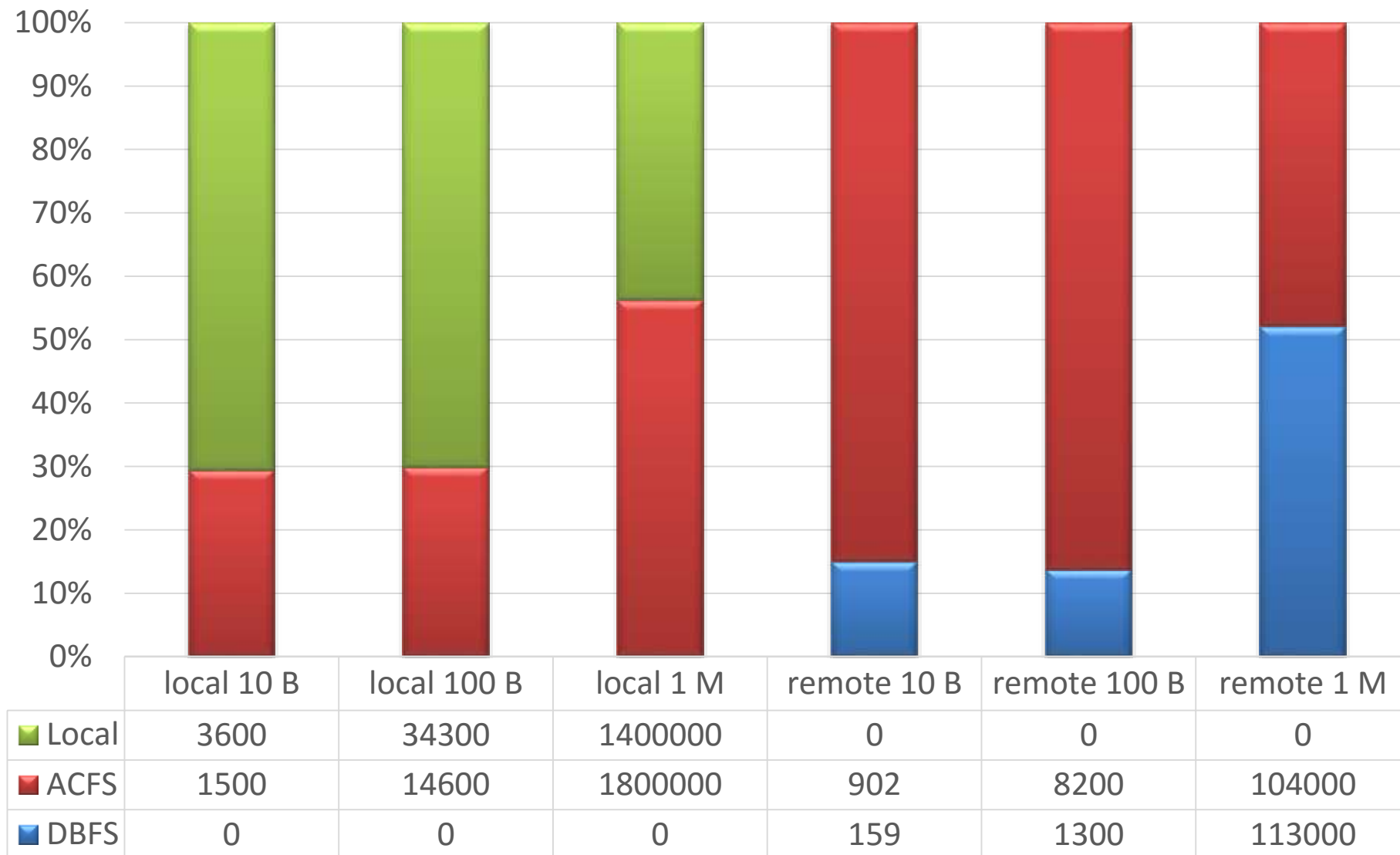
FS is unmounted and all open IOs are lost

The FS was always available for the client!

Remote Filesystems



Performance Tests



	ACFS	DBFS
Database related Files		
DB Files(redo,datafiles,...)	Not supported	Not supported
Exports(datapump)	+	-
DB logs	+	Not supported
Shared, DB and Application external tables,etc	+	+
Application Files		
Large	+	+
Small	+	-

High Available NFS (HANFS)

Supports all database files

Supported on Exadata

unlimited expansions

New ways to access DBFS:

- http/https
- ftp
- WebDAV

Based on Enterprise Manager Database Express

Any questions? ...

SecureFile is an storage option for LOBs and was introduced in 11g

LOBs can be stored as BasicFile or SecureFile

Advantages of SecureFiles:

- Compression
- Deduplication
- Encryption

```
$ sqlplus @dbfs_create_filesystem_advanced  
tablespace_name file_systemname
```

```
[compress-high|compress-medium|compress-low|nocompress]
```

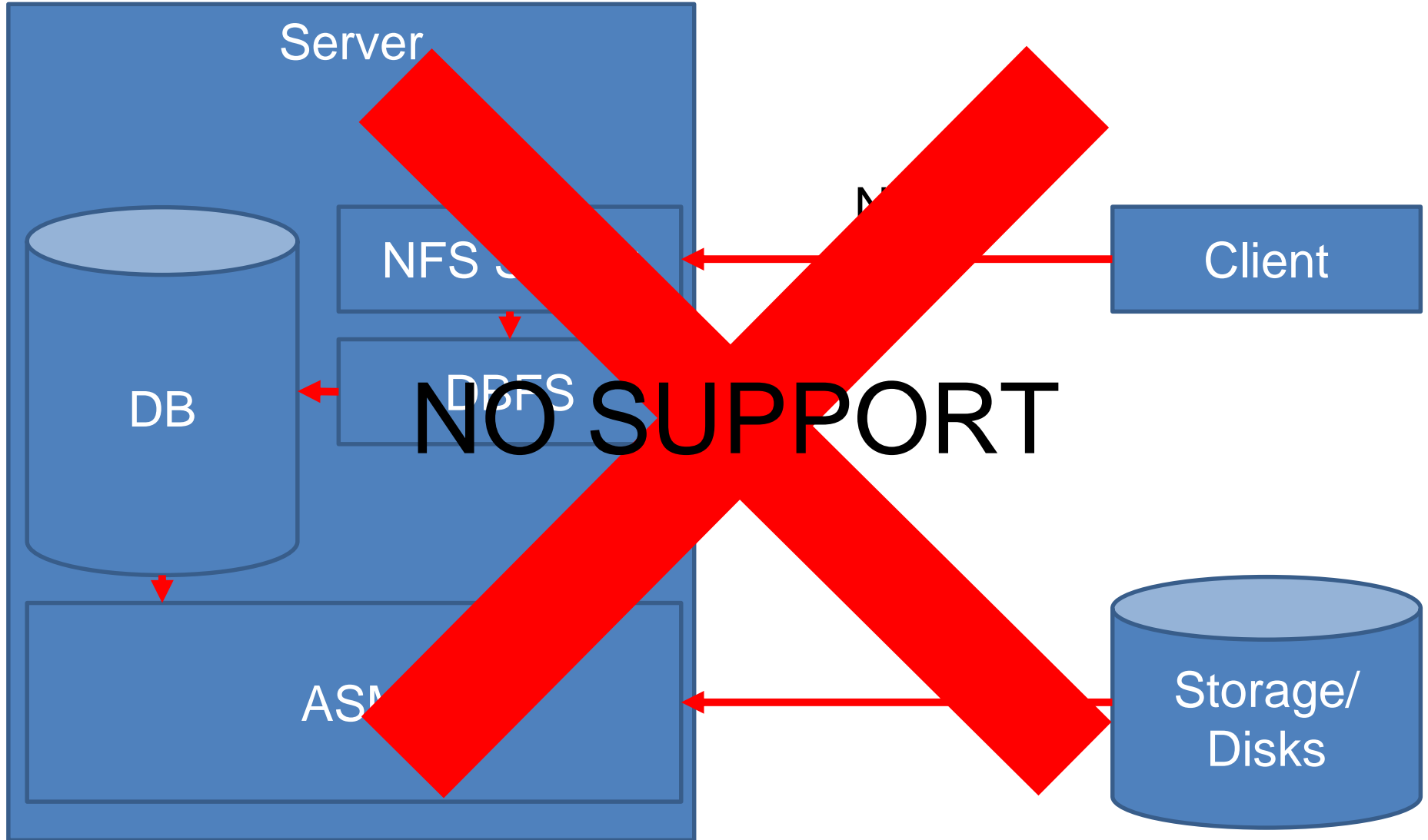
```
[deduplicate | nodeduplicate]
```

```
[encrypt | noencrypt]
```

```
[partition | non-partition]
```

The Parameter `DB_SECUREFILE` specifies whether LOBs stored as BasicFiles or SecureFiles.

Property	Description
Parameter type	String
Syntax	<code>DB_SECUREFILE = { NEVER PERMITTED ALWAYS IGNORE }</code>
Default value	<code>PERMITTED</code>
Modifiable	<code>ALTER SESSION, ALTER SYSTEM</code>



All DBFS paths must be absolute and preceded by “dbfs:”.

All commands are ls, cp, rm, mkdir.

```
$ dbfs_client db_user@db_server --command command  
[switches] [arguments]
```

```
$ dbfs_client db_user@db_server --command ls dbfs:/mydbfs
```