



PLATH

Shaping the Future of Intelligence



Oracle Secure Files

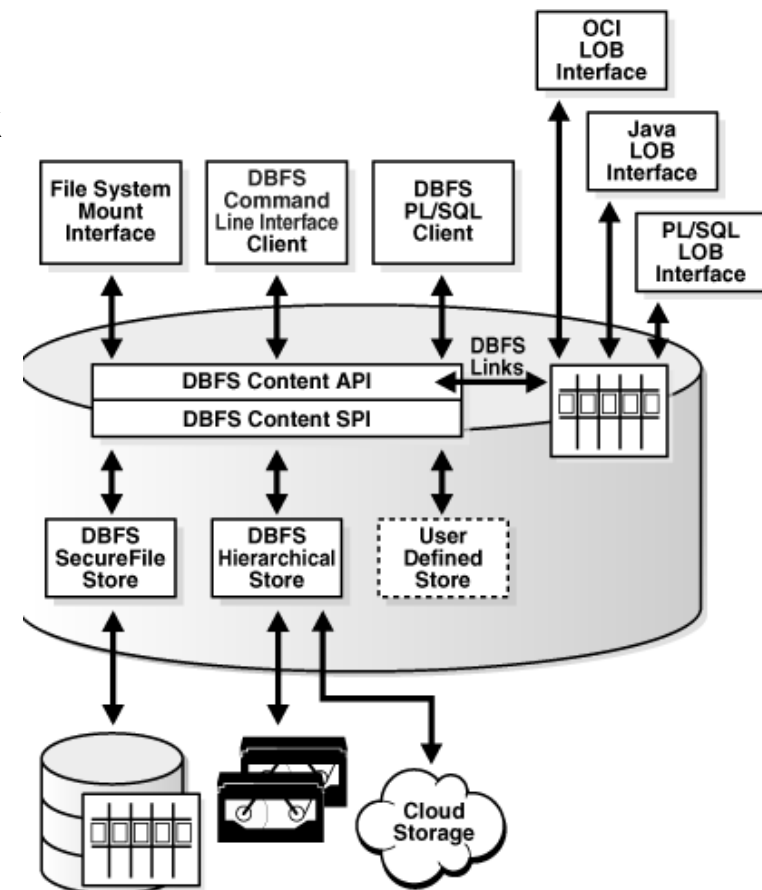
Einführung und Erfahrungen mit DBFS

- Thomas Krahn
- 14.09.2010



- Was ist SecureFiles / DBFS ?
- Installation und Konfiguration
- Benutzung
- Erste Teste und Erfahrungen
- Fazit

- Feature der Oracle Database 11g
- Dateisystem innerhalb der Datenbank
- Client – Server Trennung
- Gemeinsamer Zugriff übers Netzwerk
- Speicherung von Dateien als SecureFile LOBs
- Entwickelt für hohe Performance und Skalierbarkeit



- Was ist SecureFiles / DBFS ?
- Installation und Konfiguration
 - Server
 - Client
 - DBFS einrichten
 - Bugs und Patchsets
- Benutzung
- Erste Teste und Erfahrungen
- Fazit



- Oracle Database 11g installieren
- Tablespace für DBFS erstellen
- DBFS User erstellen und benötigte Rechte erteilen
 - DBFS_ROLE
 - CREATE SESSION
 - RESOURCE
 - CREATE VIEW
- Wallet bei Verwendung von Encryption nicht vergessen !
- DBFS DB auf allen von Oracle unterstützten Betriebssystemen möglich



- Oracle Client installieren, fertig !



- Zum mounten eines DBFS:
 - Kernel Development Package (kernel-devel) installieren
 - FUSE installieren
 - Installation des Package zur Distribution (fuse + fuse-libs) oder
 - Downloaden und Kompilieren
 - Version 2.7.3 laut Oracle Doku benutzen (2.8.3 funktioniert aber auch)
 - Gruppe „fuse“ erstellen und dem DBFS_CLIENT Benutzer zuweisen
 - Zeitsynchronisation auf Server und Clients beachten !



- sqlnet.ora einrichten
 - WALLET_LOCATION = (SOURCE = (METHOD = FILE)
(METHOD_DATA = (DIRECTORY = \$HOME/oracle/wallet)))
 - SQLNET.WALLET_OVERRIDE = TRUE ([BUG 7258404](#))

- Wallet erstellen
 - *\$ export PATH\$=\$ORACLE_HOME/bin:\$PATH*
 - Passwort für Wallet
(Buchstaben und Zahlen und min. 8 Zeichen -> PKI-01002 Error)
 - *\$ mkstore -wrl /home/oracle/wallet -create*
 - *\$ mkstore -wrl /home/oracle/wallet -createCredential
<connect_string> <user> <pass>*



- Es gibt 2 Möglichkeiten ein DBFS zu erstellen
 - Über das Betriebssystem mit den mitgelieferten Skripten
 - Über die Datenbank mittels DBFS API

- Oracle liefert 2 Skripte zur Erstellung eines DBFS mit
 - `$ORACLE_HOME/rdbms/admin/dbfs_create_filesystem.sql`
(nicht mit Standard Edition verwendbar !)
 - `$ORACLE_HOME/rdbms/admin/dbfs_create_filesystem_advanced.sql`

- Bemerkungen
 - LOB Segmente liegen im selben Tablespace wie Tabelle
 - Benutzung von SecureFile LOBs, keine Möglichkeit BasicFiles auszuwählen



- `dbfs_create_filesystem.sql`
 - Fest definierte Optionen
 - Keine Komprimierung [`nocompress`]
 - Keine Deduplizierung [`nodeduplicate`]
 - Keine Verschlüsselung [`noencrypt`]
 - Partitionierung [`partition`]
 - Ruft das Advanced Skript mit diesen Optionen auf
 - Nur mit Enterprise Edition verwendbar
(SE -> „ORA-00439 feature not enabled“)
 - Erfordert Partitioning Option



- `dbfs_create_filesystem_advanced.sql`
 - Freie Auswahl der Optionen
- Mögliche Optionen beim Erstellen eines DBFS
 - Komprimierung (Advanced Compression Option):
[`compress-high` | `compress-medium` | `compress-low` | `nocompress`]
 - Deduplizierung (Advanced Compression Option):
[`deduplicate` | `nodeduplicate`]
 - Verschlüsselung (Advanced Security Option):
[`encrypt` | `noencrypt`]
 - Partitionierung (Partitioning Option):
[`partition` | `partition-by-item` | `partition-by-path` | `partition-by-guid` | `non-partition`]



- Oracle SecureFiles benötigt in der einfachsten Nutzung keine Optionen der Oracle EE
- Damit dies auch so implementiert werden kann, sollte/muss man folgendes Script nehmen:
`$ORACLE_HOME/rdbms/admin/dbfs_create_filesystem_advanced.sql.`
- Folgende Parameter sind dabei entscheidend:
 - Nocompress und/oder nodeduplicate → keine Advanced Compression Option erforderlich
 - Noencrypt → keine Advanced Security Option erforderlich
 - Non-partitioning → Keine Partition Option erforderlich.



- DBFS API benutzen
 - Laut Oracle Doku:
Nicht als SYSDBA oder SYSOPER benutzen !
- DBMS_DBFS_SFS.CreateFileSystem()
 - Gleiche Optionen wie beim Erstellen per SQL Skript
 - Zusätzliche Angabe des LOB Tablespace möglich
 - Verwendung von BasicFiles statt SecureFiles möglich (use_bf = true)



- Inhalt des DBFS wird in 2 Tabellen gespeichert
- Partitionierung über HASH des gewählten Partitionierungsschlüssels
- 16 Partitionen bei Verwendung von Partitioning
- Dateien und Verzeichnisse sind per SELF JOIN miteinander verknüpft
- Nutzung vorhandener Tabellen mit Dateien in LOBs nicht möglich !

Column Name	ID	Pk	Null?	Data Type
VALID	1	N	N	NUMBER
CSNAP#	2	N	N	NUMBER
LSNAP#	3	Y	N	NUMBER
PATHNAME	4	N	N	VARCHAR2 (1024 Byte)
ITEM	5	N	N	VARCHAR2 (256 Byte)
PATHTYPE	6	N	N	INTEGER
FILEDATA	7	Y	N	BLOB
POSIX_NLINK	8	Y	N	INTEGER
POSIX_MODE	9	Y	N	INTEGER
POSIX_UID	10	Y	N	INTEGER
POSIX_GID	11	Y	N	INTEGER
STD_ACCESS_TIME	12	N	N	TIMESTAMP(6)
STD_ACL	13	Y	N	VARCHAR2 (1024 Byte)
STD_CHANGE_TIME	14	N	N	TIMESTAMP(6)
STD_CONTENT_TYPE	15	Y	N	VARCHAR2 (1024 Byte)
STD_CREATION_TIME	16	N	N	TIMESTAMP(6)
STD_DELETED	17	N	N	INTEGER
STD_GUID	18	N	N	INTEGER
STD_MODIFICATION_TIME	19	N	N	TIMESTAMP(6)
STD_OWNER	20	Y	N	VARCHAR2 (32 Byte)
STD_PARENT_GUID	21	N	N	INTEGER
STD_REFERENT	22	Y	N	VARCHAR2 (1024 Byte)
OPT_HASH_TYPE	23	Y	N	VARCHAR2 (32 Byte)
OPT_HASH_VALUE	24	Y	N	VARCHAR2 (128 Byte)
OPT_LOCK_COUNT	25	Y	N	INTEGER
OPT_LOCK_DATA	26	Y	N	VARCHAR2 (128 Byte)
OPT_LOCK_STATUS	27	Y	N	INTEGER



- Update des Dateinamen im PATHNAME ändert Dateinamen
- Update des ITEM ändert Dateinamen nicht
- Anscheinend kein Bezug zwischen PATHNAME und ITEM

VOLID	CSNAP#	LSNAP#	PATHNAME	ITEM	PATHTYPE	FILEDATA
0	0		/linux-2.6.35.2/Documentation/filesystems/caching/fscache.txt	fscache.txt	1	
0	0		/linux-2.6.35.2/Documentation/cgroups	cgroups	2	
0	0		/linux-2.6.35.2/Documentation/cgroups/cpuacct.XTX	cpuacct.txt	1	
0	0		/linux-2.6.35.2/Documentation/console	console	2	
0	0		/linux-2.6.35.2/Documentation/cris	cris	2	
0	0		/linux-2.6.35.2/Documentation/hwmon	hwmon	2	
0	0		/linux-2.6.35.2/Documentation/hwmon/lm77	lm77	1	

- Vorsichtig beim manuellen Ändern von PATHNAME

```

oracle@oel-r5-tnr:/mnt/dbfs/KERNELFS/linux-2.6.35.2/Documentation/cgroups
-rw-r--r-- 1 oracle oinstall 1941 Aug 13 22:44 cpuacct.txt
?----- ? ? ? ? ? cpuacct.XTX
-rw-r--r-- 1 oracle oinstall 36719 Aug 13 22:44 cpuset.txt
-rw-r--r-- 1 oracle oinstall 1898 Aug 13 22:44 devices.txt
-rw-r--r-- 1 oracle oinstall 4134 Aug 13 22:44 freezer-subsystem.txt
-rw-r--r-- 1 oracle oinstall 14133 Aug 13 22:44 memcg_test.txt
-rw-r--r-- 1 oracle oinstall 27354 Aug 13 22:44 memory.txt
-rw-r--r-- 1 oracle oinstall 6170 Aug 13 22:44 resource_counter.txt
[oracle@oel-r5-tnr cgroups]$
  
```



- Bisher 3 öffentliche Bugs bei Metalink zu finden
 - Bug 8733225 - slow performance on DBFS [ID 8733225.8]
 - Bug 9367140 - chgrp on a file of a DBFS mounted file system corrupts the owner [ID 9367140.8]
 - Bug 9309554 - Errors when mounting a DBFS store using fstab due to missing libraries
- Aktuell 2 Patches für dbfs_client in DB Version 11.2.0.1
 - [Patch:9678030](#), [Patch:9526078](#)
- Aktuell 3 Patches für Exadata Kunden
 - [Patch:9914093](#), [Patch:9655912](#), [Patch:9526078](#)

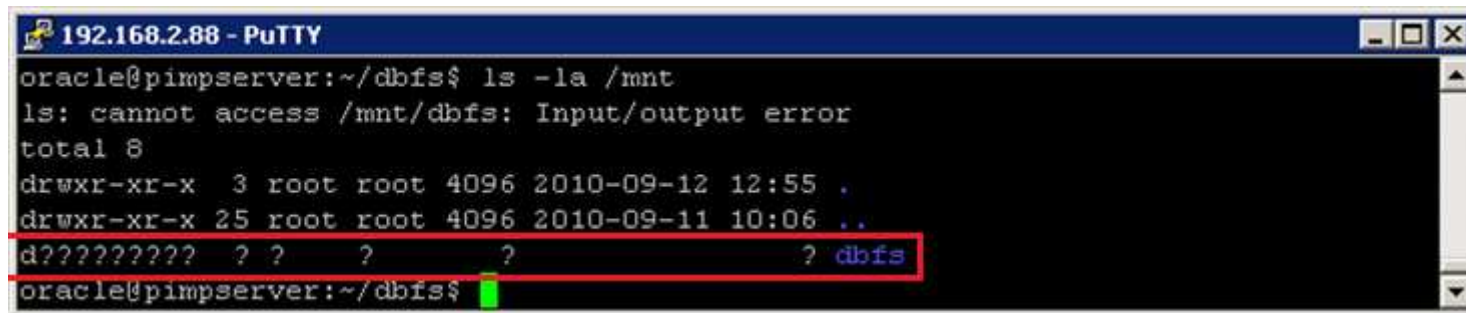


- Was ist SecureFiles / DBFS ?
- Installation und Konfiguration
- Benutzung
 - Mounten
 - DBFS_CLIENT
- Erste Teste und Erfahrungen
- Fazit

- DBFS kann als Dateisystem nur in Linux gemountet werden
- Benutzung von Linux Standard Befehlen möglich
(*cp, mv, rm, ls, mkdir, rmdir, chown, chmod, du, ...*)
- \$ORACLE_HOME und \$LD_LIBRARY_PATH müssen gesetzt sein !
- Klassisches User/Pass Verfahren
 - *\$ dbfs_client <user>@<db> -o options <mount_point>*
 - Prozess bleibt bis zum unmounten im Vordergrund !
- Wallet Benutzung
 - *\$ dbfs_client /@<db> -o options <mount_point>*
- Eintrag in /etc/fstab



- Mount-Verzeichnis bei Verwendung von Wallets unbrauchbar wenn User/Pass nicht korrekt sind



```
192.168.2.88 - PuTTY
oracle@pimpserver:~/dbfs$ ls -la /mnt
ls: cannot access /mnt/dbfs: Input/output error
total 8
drwxr-xr-x  3 root root 4096 2010-09-12 12:55 .
drwxr-xr-x 25 root root 4096 2010-09-11 10:06 ..
d????????? ? ? ? ? ? dbfs
oracle@pimpserver:~/dbfs$
```

- Mount-Verzeichnis muss zum mounten dem User gehören der mountet, nach dem mounten gehört es jedoch wieder *root* !
- Unmounten mittels *fusermount* statt *umount*
 - `$ fusermount -u <mount_point>`



- Der Befehl *dbfs_client* kann für einzelne Dateioperationen genutzt werden
- Klassisches User/Pass verfahren
 - `$ dbfs_client <user>/<pass> @<db> - - command <command>`
`dbfs:/<store_name>`
 - DIRECT_IO noch nicht unterstützt !
- Wallet benutzen
 - `dbfs_client /@<db> - - command <command> dbfs:/<store_name>`
- Begrenzte Möglichkeiten
 - `mkdir, rmdir, cp, rm, ls`
 - Bisher kein `mv` implementiert !



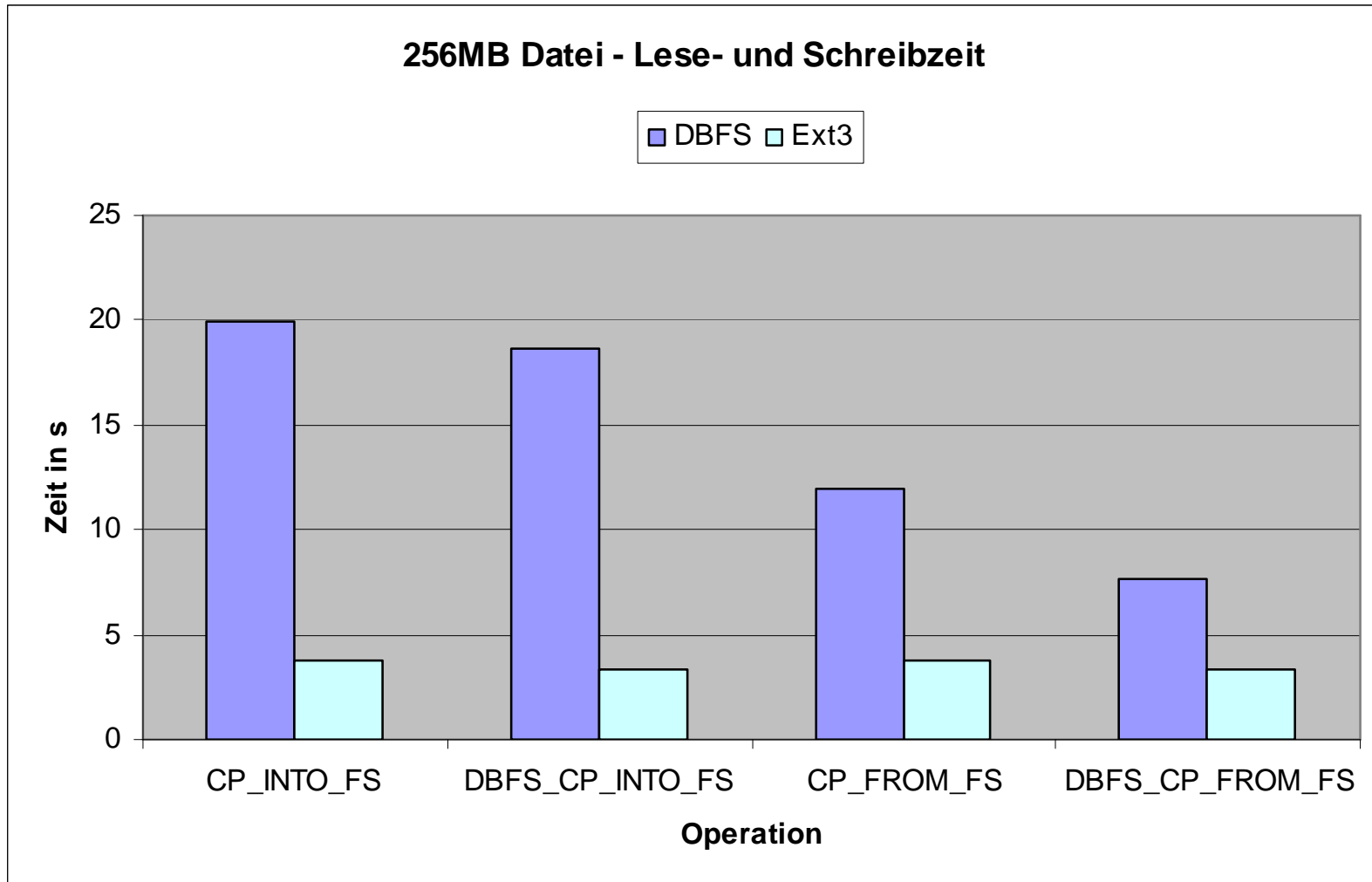
- Was ist SecureFiles / DBFS ?
- Installation und Konfiguration
- Benutzung
- Erste Tests und Erfahrungen
 - Mountet DBFS vs. DBFS_Client
 - Kernel Test
- Fazit

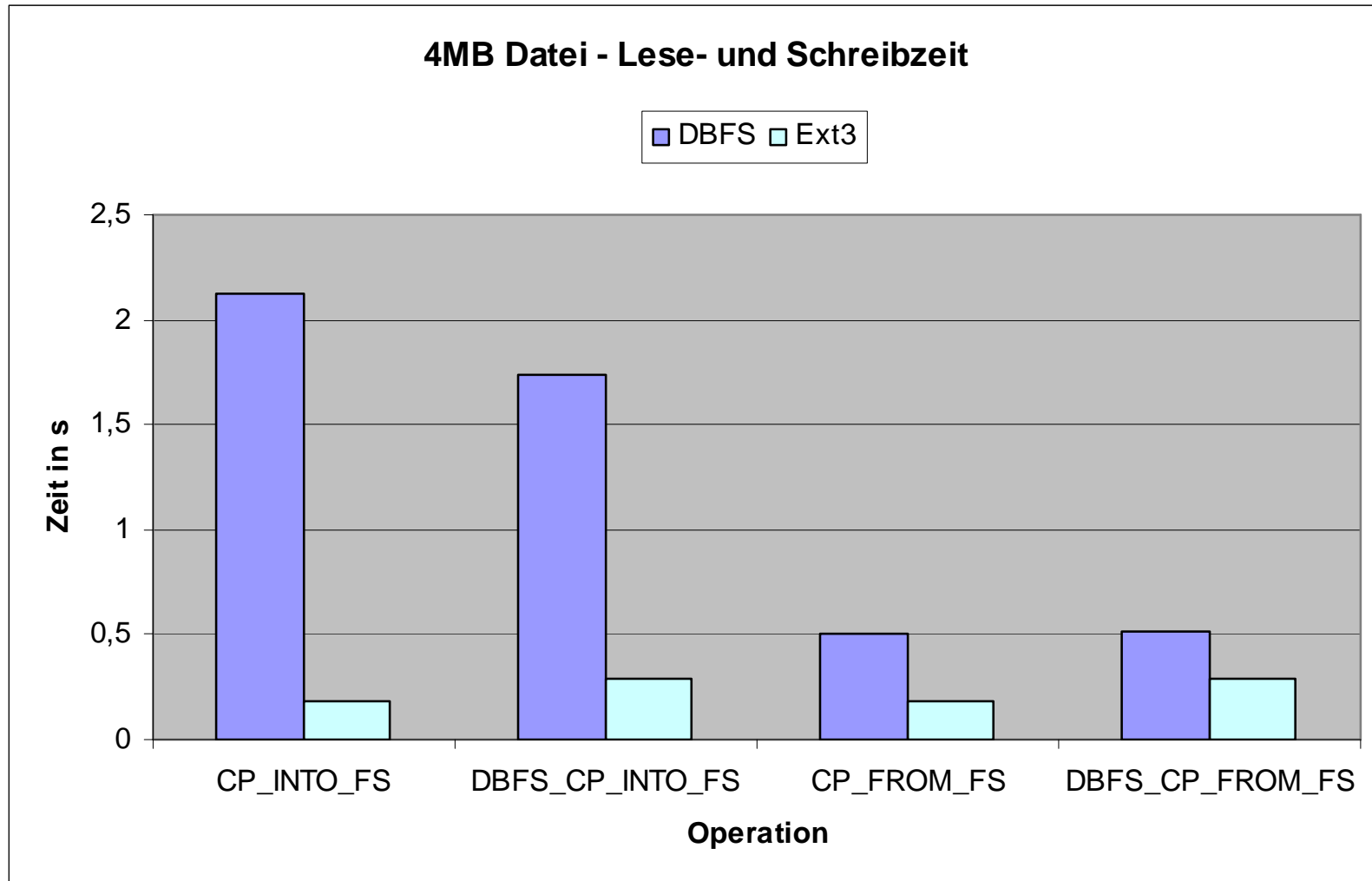
- Oracle Enterprise Linux 5.5 in Vmware VM
- 4x Xeon 3,2Ghz CPU
- 16GB Ram
- 2x146GB HDD im RAID1
- Oracle Database 11g R2 64Bit (11.2.0.1) Single Instance
 - 8 GB MEMORY_MAX_TARGET
 - 4 GB DB Buffer Cache
 - ASM
 - DBFS nocompress, nodeduplicate, noencrypt, non-partition
- FUSE 2.8.4

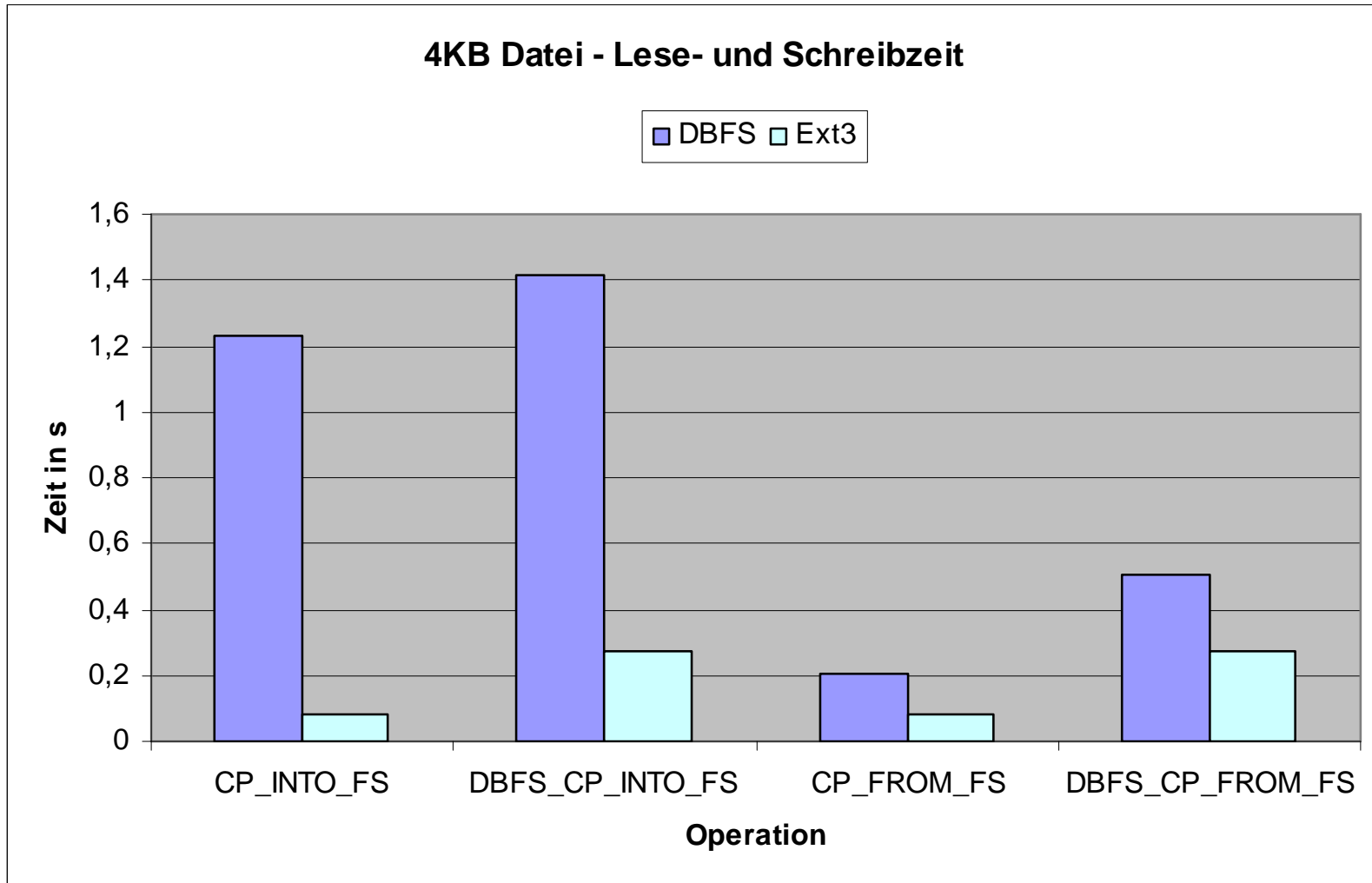


- Ermittelt wird die benötigte Zeit zum Kopieren , Umbenennen und Löschen von Dateien der Größe 4KB, 4MB, 256MB in das gemountete DBFS
 - Datei in das DBFS kopieren
 - Datei innerhalb des DBFS umbenennen
 - Datei aus dem DBFS auf lokalen Storage kopieren
 - Datei aus dem DBFS löschen
- Der Test wurde für jede Dateigröße 100 gestartet um einen realistischen Mittelwert zu bekommen
- Die Zeiten wurden unter Verwendung von Linux Befehlen sowie DBFS_Client getestet









- Ermittelt wird die benötigte Zeit zum entpacken und packen des Linux Kernel 2.6.35 auf ein gemountetes DBFS
- Zum Vergleich wurde der Test zusätzlich auf dem selben Storage mit ext3 Dateisystem ausgeführt
- Server – Server – Test
- Verwendung des Loopback Interface
 - `$ tar -xjf linux-2.6.35.tar.bz2 /mnt/dbfs/TESTFS`



Aktion	DBFS Time	Ext3 Time
Entpacken	1h 52min	23s
Packen	51min	58s
Löschen	47min	2s

- DBFS_CLIENT Prozess während dieser Zeit bei 60% - 85% CPU Last
- DBWR unter hoher last
- Trace der Session zeigt COMMIT nach jeder Datei / jedem Verzeichnis



- DBFS als Dateisystem durchaus brauchbar
- Schnelle und einfache Installation
- Einfache Sicherung und Wiederherstellung mit RMAN
- Transfer eines Dateisystems mittels Transportable Tablespace sehr einfach
- Verwendung von FLASHBACK ermöglicht einen Blick in die Vergangenheit von Dateiinhalten
- Replizierung eines DBFS mit Dataguard möglich
- Hoch verfügbares Dateisystem im Real Application Cluster
- Keine PL/SQL Packages mehr nötig um Dateien in die DB zu transferieren



- DBFS_Client nur in Linux Systemen verfügbar
- Hohe CPU Last am Client beim sequentiellen Schreiben vieler kleiner Dateien
- Geringe Performance beim Schreiben vieler kleiner Dateien
- Keine Möglichkeit vorhandene Datei-Tabellen als DBFS zu nutzen
- Keine freie Wahl des Partitionierungsschlüssels



```
oracle@oel-r5-tkr:/opt/oracle/diag/rdbms/orcl/ORCL/trace
SQL> select dbms_dbfs_sfs.getversion('KERNELFS') from dual;

DBMS_DBFS_SFS.GETVERSION('KERNELFS')
-----
0.5.0

SQL>
```







- FUSE: <http://fuse.sourceforge.net>
- DBFS Installation:
http://download.oracle.com/docs/cd/E11882_01/appdev.112/e10645/adlob_client.htm
- DBFS Performance Guide:
<http://www.oracle.com/technetwork/database/features/performance/securefilesperformancepaper-130360.pdf>





PLATH

Gotenstraße 18
20097 Hamburg
Germany
www.plath.de

Tel.: +49 40 23734-213
Fax: +49 40 23734-173
Email: info@plath.de

